Aalto University

School of Science

Master's Programme in Computer, Communication and Information Sciences

Valtteri Wahlroos

Design and implementation of a gameplay UI for a MOBA mobile game

Master's Thesis

Espoo, April 21, 2018

Supervisor: Hämäläinen, Perttu, Professor

Thesis advisor(s): Lukka, Lauri, M.A.

Aalto University
School of Science

Abstract of master's thesis

Author Valtteri Wahlroos

Title of thesis Design and implementation of a gameplay UI for a MOBA mobile game

Master's programme Computer, Communication and Information Sciences

Thesis supervisor Perttu Hämäläinen

Major or Minor/Code ARTS3049

Department Department of Computer Science

Thesis advisor(s) Lauri Lukka, M.A.

Date 21.04.2018 **Number of pages** 71 **Language** English

Abstract

In this master's thesis the author was developing a multiplayer online battle arena (MOBA) mobile game with a team as an employee in Rovio Entertainment Corporation. The aim of the project was to design and develop a MOBA mobile game user interface (UI). The project was defined as successful if test users were able to play without instructions in the beginning or middle of gameplay and they did not need any help in executing actions or using the user interface during the gameplay.

A further aim was to design and develop a MOBA mobile game which would be played as a one-handed game in portrait mode. One-handed portrait gameplay was wanted to create casual gameplay. The first problem in the portrait mode was that playing with one hand was slower than with two hands, and fast actions and reactions impacted winning possibilities. In user testing, players played the game two-handed allowing them to alter between actions faster and therefore giving them an advantage over other players. Preventing the usage of two hands in any circumstances had effects on gameplay making it unenjoyable. Tap-to-move and tap-to-attack mechanics could have possibly fixed the problem. This possibility wasn't tried by the team because it could have had negative effects to the game feel and immersion. For these reasons the game was modified to support landscape mode, allowing two-handed gameplay.

In order to create a unique combat system that stands out among other MOBA mobile games in landscape mode, the UI functionality was made more complex which confused players. In traditional MOBA mobile games the player's actions are performed by tapping and optionally by dragging the action buttons. In the team's game, instead of tapping and dragging action buttons, the player chose the active weapon from two weapons and always used the same button for aiming and shooting regardless of the active weapon. In addition, the weapons had infinity ammo capacity but clip sizes were finite. For this reason, when active weapon's clip was empty, players had to reload or change the weapon. However, players did not understand how weapon switching or ammunition worked. After iterating the game to work more similarly as traditional MOBA mobile games, players learned the game mechanics faster. In conclusion, it was found that it is hard to break established UI conventions without sacrificing the intuitiveness of the interface.

Keywords design, implementation, UI, MOBA, mobile games, games

Aalto-yliopisto
Perustieteiden korkeakoulu

Diplomityön tiivistelmä

Tekijä Valtteri Wahlroos

Työn nimi Käyttöliittymän suunnittelu ja toteutus taisteluareenamoninpelin pelitilaan mobiililaitteissa

Koulutusohjelma Tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelma

Valvoja Perttu Hämäläinen

Pää tai sivuaine/koodi ARTS3049

Työn ohjaaja(t) Lauri Lukka, Taiteiden maisteri

Päivämäärä 21.04.2018 **Sivumäärä** 71 **Kieli** Englanti

Tiivistelmä

Tässä diplomityössä kirjoittaja kehitti taisteluareenamoninpeliä (multiplayer online battle arena, MOBA) mobiililaitteille tiiminsä kanssa työntekijänä yrityksessä Rovio Entertainment Corporation. Projektin tavoitteena oli suunnitella ja kehittää käyttöliittymä (user interface, UI) MOBA mobiilipelin pelitilaan. Projektin onnistumiskriteereinä oli, että testikäyttäjät pystyvät pelaamaan ilman ohjeita pelin alussa tai sen aikana ja he eivät tarvitse apua toimintojen tekemiseen tai käyttöliittymän käyttämiseen pelitilassa.

Tavoite oli myös tehdä MOBA mobiilipeli, jota voi pelata puhelin pystyasennossa yhdellä kädellä. Yhden käden peli pystykuvatilassa oli haluttu, jotta peli soveltuisi useampiin arkielämän tilanteisiin. Ensimmäinen ongelma pystykuvatilassa oli, että pelattavuus yhdellä kädellä oli hitaampaa kuin kahdella kädellä. Nopeat toimintojen käyttämiset ja reaktiot vaikuttivat voittomahdollisuuksiin. Käyttäjätesteissä pelaajat pelasivat peliä kahdella kädellä, jolloin he pystyivät vaihtelemaan toimintoja nopeammin ja pärjäsivät näin paremmin kuin muut pelaajat. Kahden käden käytön estäminen kaikissa olosuhteissa teki pelaamisesta epämukavaa. Hyökkäämisen ja liikkumisen salliminen kohdetta napauttamalla olisi saattanut korjata edellä mainitun ongelman. Kehittäjätiimi ei halunnut muuttaa peliä toimimaan napauttamalla, jotta peliin uppoutuminen ei kärsisi. Näistä syistä johtuen peli muutettiin toimimaan näyttö vaakasuorassa.

Jotta taistelujärjestelmä erottuisi muista MOBA mobiilipeleistä, käyttöliittymä tehtiin monimutkaisemmaksi, mikä puolestaan hämmensi pelaajia. Perinteisissä MOBA mobiilipeleissä pelaajan toiminnot suoritetaan napauttamalla ja mahdollisesti raahaamalla toimintapainikkeita. Kehittäjätiimin pelissä pelaaja sen sijaan valitsi aktiivisen aseen kahden eri aseen välillä ja käytti aina samaa painiketta tähtäämiseen ja ampumiseen riippumatta siitä, mikä ase on aktiivinen. Tämän lisäksi aseiden ammusten määrä oli loputon, mutta aseiden lippaiden koko oli rajallinen. Tästä syystä pelaajien tuli ladata tai vaihtaa asetta, kun aseen lipas oli tyhjä. Pelaajat eivät kuitenkaan ymmärtäneet kuinka aseiden vaihtaminen ja ammusten hallinta toimivat. Kun peliä iteroitiin toimimaan enemmän perinteisten MOBA mobiilipelien kaltaisesti, pelaajat oppivat pelin mekaniikat nopeammin.

Avainsanat suunnittelu, toteutus, käyttöliittymä, MOBA, mobiilipelit, pelit

Table of contents

1 Introduction

The purpose of this master's thesis was to develop and optimize a multiplayer online battle arena (MOBA) mobile game in a way that new players could learn how to play the game immediately. The main aim in the project was to design and develop a user interface (UI) for gameplay in a MOBA mobile game. A further aim was to design and develop a MOBA mobile game that could be played as a one-handed game in portrait mode. One-handed portrait gameplay was desired in order to create a casual form of gameplay.

At the start of this project, the author was given a prototype of the game with limited actions and UI. A player was able to move in the direction of the player's input, use melee attack and throw a ball in the player's chosen direction. Also, potential ideas for new actions and UI design were given at the beginning of the project.

The goal of the UI design was to provide an easy-to-understand and easy-to-use gameplay interaction. Many alternative UIs were made and tested with test users. The UI was defined to be successful if test users were able to play without instructions in the beginning or middle of gameplay and they did not need any help in performing actions or using the user interface during the gameplay. In addition to the UI, some aspects of user experience (UX) were analyzed. Organized test sessions were primarily focused on UI but also UX aspects were analyzed.

There was a possibility that the UI for the game was not successful in portrait mode. In that case, the game would be designed in landscape mode. The first anticipated problem areas were how the game works with different screen sizes and do the fingers and buttons cover too much of the gameplay area.

The scope of the UI in this master's thesis is limited to the UI that players use during core gameplay. Therefore, menus and other irrelevant UI buttons such as the pause menu button were left outside of the thesis. The scope of UX is mainly limited to the fun and enjoyability factors in the core gameplay.

The author took part in developing the game together with a team as an employee of Rovio Entertainment Corporation.

2 Definitions

This master's thesis is focused on MOBA games, mobile games, user interface and user experience. To design user interface and user experience for a mobile MOBA game, these terms require clear definitions to frame the subsequent analysis.

2.1 User interface

This master's thesis heavily refers to user interface, commonly abbreviated UI. I begin by giving a definition of UI and what is it used for.

User interface in a game is a system that provides relevant information about the gameplay and it gives the right tools for interacting with the game. Any feature that provides information or assists the player in interacting with the game is part of the user interface. These include all the controllers and screen and software features in the game, in addition to the audiovisual features. Graphical user interface (GUI) is part of the UI and it covers all the graphical aspects. Head-up display (HUD) is an information display technology that projects gameplay information on the screen and it is a part of GUI. Traditional GUI content in action games are the minimap, health indicator, and weapon selection indicator.[1]

Some parts of the UI can be integrated in the game world and a part of them superimposed as overlay. Examples of these kinds of UI features can be, respectively, visual effects on climbable towers or icons and menus in HUD, which are against the illusion of coherent natural world. Players accept superimposed UI features as long as the information is relevant and sufficient in a given situation. The UI has to communicate the information to the player in a clear and consistent manner. Going beyond that may have the negative result of annoying players. [1]

Diegesis is a concept used traditionally in literary and in film theory and it has been lately applied to games [2]. Diegesis is used in film to describe the world where the character is located and non-diegesis refers to all the features and elements that only the audience is able to see or hear such as titles and musical score. Similarly, in games the UI consists of both diegetic and non-diegetic elements. Diegetic is the narrative of the game. It contains everything that the player character can see, hear, or interact with. Non-diegetic elements are only visible to the player and these elements are, for example, HUD and all of its parts such as the health bar and minimap. These non-diegetic elements make the game more accessible by providing additional information about the game world, other game characters, the status of the player, and guide how to interact with items. [3] In the game industry, it is assumed that adding non-diegetic features to the game makes the game less immersive [4]. However, some non-diegetic elements such as the right kind of music for certain situations can increase tension and excitement in the overall user experience [3]. Therefore, music is an effective and powerful way of creating emotions [5]. Furthermore, monitoring additional information on the UI can break the immersion of the game world and therefore negatively impact user experience. However, while novice players are more likely to be in need of non-diegetic elements to be able to progress in the game, expert players may have a better gaming experience when there are no non-diegetic elements. [3]

Stein C. Llanos studied in his PhD project how the switched off HUD in Assassin's Creed affected the gameplay experience. Some players felt the game much more harder and got stuck easier because all the indicators like objective indicators were missing. Some players took the missing HUD as a positive change that made the game more realistic and their focus was much more on the (non-player characters) NPCs and the environment of the game. [1]

Another research group studied how the missing non-diegetic elements affect player involvement in games. In the first study, nine participants played Level 3: Uprising in Battlefield 3 with and without non-diegetic elements. The differences of diegetic and non-diegetic versions of the game are shown in **Figure 1**. In the diegetic version of the game, the following features were removed from the user interface: crosshairs, teammate markers and names, ammo display, compass, goal markers, grenade indicators, notifications such as saving, visual objectives, item walkovers and environment interaction. The results for diegetic and non-diegetic versions did not vary significantly in terms of enjoyment, frustration or the CEGE questionnaire. However, in the interviews some players noted that in the diegetic version they had to think more and be more attentive to the environment, which increased involvement in the game. The findings pointed out that removing non-diegetic elements won't create significant barriers to engagement. Furthermore, by the interviews it was suggested that the diegetic interface

Figure 1. Diegetic (above) vs non-diegetic (below) gameplay screens. [3]

can increase the level of involvement in and challenge of the game by requiring more attention from player. [3]

From the same research group, 24 players (divided into novice and expert players) played the same game as in the last study described above. However, this time the participants were randomly placed to play only one of the game versions, diegetic or non-diegetic. This study indicated that there is a difference in the immersion between diegetic and non-diegetic interfaces. However, this difference was only seen with expert players. With expert players the diegetic interface provided more immersion to the game than a non-diegetic interface. [3]

Even though the UI should give relevant and sufficient information to allow players to interact meaningfully with game mechanics and the game world, different players have different thoughts about what is meaningful. For designing an UI, it is important to know what kind of game experience needs to be created and to recognize which features are actually important to be shown to the player. A minimal UI that seems to be natural for the game may not be the best way to give critical information nor be effective if it needs to be gauged continuously. [1]

For designing a user interface for a mobile game, the features of the end user's device have to also be considered. For example, an iPhone X has a couple of major aspects requiring attention. An image of an iPhone X can be seen in **Figure 2**. First, the iPhone X has rounded corners in the display. This means that by having UI elements like buttons near to any of the four corners, these buttons may be only partially visible in the display of the iPhone X. The second aspect is that the display has a 'notch' on the top. This notch breaks the symmetrical shape of the display by having two branches on both sides of the notch. This notch has to be considered in game design for the bottom side of the screen as well, because the phone can be rotated upside down. The final major feature to be taken into account is the new changed swipeable 'home button'. By removing the physical

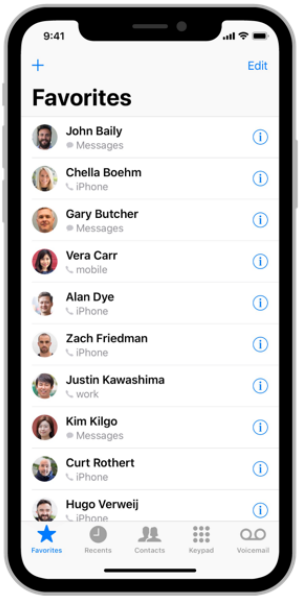

Figure 2. iPhone X. At the bottom of the screen is the new swipeable 'home button' shown as a black narrow line. [6]

home button from the iPhone X, there is now a narrow black line at the bottom of the screen depending on the orientation of the phone at any moment. By swiping up on this narrow line, the user gets back to the home screen. This feature has two issues with implications on game design. First, the player should not need to use the swipe feature at the middle of the bottom side of the display in any situation during the game. Otherwise, the player may accidentally close the game. Secondly, the narrow black line will cover a very small area of the middle bottom side of the screen; thus, there should not be anything that could be covered all the time. [6] These kinds of items could be UI buttons that have text or an image that would be covered by the black line.

Mobile games can be controlled through a user interface in many different ways: for example, through virtual buttons, tilting, swiping and even voice commands. These kinds of games are, for example, Arena of Valor, Classic Labyrinth 3d Maze, Fruit Ninja, and Chicken Scream, respectively. [7-10] Arena of Valor is a MOBA game and it is played in landscape mode. A player can move the hero by using a virtual joystick button on the left bottom corner of the screen. Other virtual buttons and joysticks will be used by the player to buy items, select abilities, and aim in the shooting direction. [7] In Classic Labyrinth, the player tilts the phone to tilt the platform where the ball and maze are located. By tilting the platform, the ball starts to roll in the tilted direction. [8] In Fruit Ninja, the player's objective is to destroy fruits with a sword by swiping over the display and hitting the fruits [9]. Chicken Scream is a 2D platform game in which the player's objective is to get the chicken to the goal by avoiding gaps and other malicious objects. This game, however, is not played by virtual buttons/joysticks, tilting, or swiping—but with voice commands. When the player keeps making a sound, the chicken walks forward. When the player makes a louder sound, the chicken will jump. The player has to use these two types of sound to get to the goal. [10]

2.2 User experience

Games are interactive entertainment played to experience something interesting and often also emotional, be that emotion joy, companionship or even grief [5,11]. In addition to study games in the field of human-computer interactions (HCI), there is also need to analyze from another perspective called user experience [11].

Video games have become the fastest growing form of human recreation and the annual revenues from video games have surpassed those of Hollywood. Video games occupy an increasing large part of people's time and energy. According to one theory, players can be divided into four classes: killers, achievers, socializers and explorers. Killers wants to act on other players, such as by killing them. Achievers want to act on the game world. Socializers wants to interact with other players. Explorers wants to interact with the game world through exploration and manipulation. The theory states that all of these player types have to receive gratification for the game to be successful. [12]

Modern digital games are experiences that combine almost all forms of entertainment and art through videos, 3D animations, and sounds. The core of digital games is the interaction between the player and the game. One of the signs of success in game design is considered to be simple and easy rules that create interesting and unpredictable dynamics, multidimensional aesthetics, and emotional experiences. For example, an easy and simple rule is where a player can pick items to pocket and drop them on the ground, which can

create donation and trade dynamics along with social interaction by helping others and creating friendships. Since game designers and developers cannot force players to behave in a desired way and successful game design needs to predict player behavior, game design incorporates motivational and behavioral parts of psychology. [13]

Psychology divides motivation into internal and external subparts. Internal motivation refers to psychological basic needs that motivate actions with external rewards. Optimizing rewards and challenges as part of the external motivation is only a part of a good game design. Having the game based purely on external rewards may have negative effects. Losing a battle repeatedly decreases one's sense of ability, which in turn decreases motivation in the long run. One's sense of ability is the keystone of internal motivation and supporting the internal motivation is seen as good basis for game design. However, creating internal motivation in a game can be hard because it is hard to know what players think, feel, or desire. [13]

A common way of studying user experience in games is to look at the fun factor of games. Fun lead to enjoyable gameplay and makes the players want to play the game more. To create more enjoyable, interesting and satisfying games (or computer systems), challenge, fantasy, and curiosity were also studied. After these, many other heuristics have been introduced to help design better games. These heuristics are also good indicators of aspects that should be included or modified in the game. Nevertheless, evaluating such games requires a psychology-based approach. [11] Other theoretical frameworks such as Mechanics, Dynamics and Aesthetics (MDA) try to integrate game structures to experiential outcomes [14]. Theoretical frameworks and heuristics share the same problem as they are validated by professional game developers and designers. However, theoretical models are needed to create a base and a starting point for the study where the phycological approach can continue to find the core of the UX. [11]

The main motivation for playing is considered to be the level of fun in the game, which is why the source of fun in games has been researched. [11] In Lazzaro's empirical study, the motivation to play digital games consisting of 30 players and 15 non-players were studied. Based on this study, four key features were found. These four key features were internal sensations, challenge, game's ability to fully absorb players' attention, and social experiences. Internal sensations are all the feelings that the player handles during the gameplay such as excitement and relief. Challenge gives the players opportunities for harder gameplay, solving problems, and to develop strategies. Games that received attention throw the players into exciting adventures to generate emotions and experiences such as awe and mystery. Social experience covers different gameplay situations between players such as teamwork, competition, social bonding, and personal recognition. [15]

Another similar study was conducted with a larger number of participants (n = 550). It included the dimension of diversion from other things such as studying. To understand player motivation, even larger empirical studies have been held with 3200 participants. Even though this study was made for only one genre (massively multiplayer online role-playing games, MMORPG), the results were similar to previously mentioned cross-game studies. However, studies with a single genre give more accurate results: for example, challenges in the games such as competition and advancement. [11]

The results of game studies on identifying the components of fun can be categorized into emotions and internal sensations, challenge, fantasy, narrative, curiosity, and social interaction. In addition to these, wider concepts such as immersion, presence, flow, involvement, engrossment, and engagement are considered to be part of the fun of and UX in games. [11]

The motivation for trying and seeing new art and entertainment is interest. Interest is a function that motivates people to learn and explore new things. When people are motivated to learn for its own sake, interest will contribute to the development of knowledge, skills, and experience. While being interested, session times with different forms of art increase, people are more immersed with the object, and their learning curves are steepened. When dealing with boring tasks, new strategies or complex settings are created to maintain interest. However, finding ways to create interest in many people can run into two problems. First, people have different areas of interest. A single person's point of interest may not align with the points of interest of other people. Second, the level of the interest for the same person differs over time. A once-interesting game can become boring, frustrating or aversive. Interest motivates to learn about something new and complex, but once people understand the thing, it will no longer be interesting. [16]

In addition to the fun factor, there are also other aspects that motivate players to play games. The first of these is autonomy. When a player undertakes activities out of interest or personal value, perceived autonomy is high. Choices, rewards as information feedback, and non-controlling instructions increase the sense of autonomy and therefore also the core motivation. Another source of motivation is competence. Competence can be made of challenges or from the feeling of influence. Opportunities to acquire new skills or abilities are factors that enhance the experience of competence. Optimizing the challenge of these opportunities will increase the perceived competence and the motivation of gaming. Also, intuitive controls can have an impact on motivation by allowing a greater sense of freedom and thus enhancing competence. Another source of motivation is relatedness. Relatedness refers to experiences of interacting with other real players in multiplayer games. Nevertheless, it is possible that relatedness is also experienced with computer-generated personalities and artificial intelligent (AI) players. [12]

At the GDC conference in 2013, Jenova Chen described how they designed Journey. Games can be innovated and built around new technology, art style, mechanics, or emotions, and Chen's team wanted to go with emotions by their personal preferences. Different emotions vary in quantity, and Chen's examples were fun, accomplishment, empowerment, and feeling of socializing. Even though socializing is usually thought to stem from interacting with other players, it can also take place with AI characters such as in The Sims. [5]

In the development of Journey, Chen and his team wanted to create the feeling of interacting with other players in online gaming. He mentioned the basic problems of many online games, which is that players do not care about others, the main topic of discussion is loot, and players think about how they can apply their powers against others. To create a more meaningful mode of player interaction, they chose to make the player less powerful against monsters, remove all distraction and noises of war by not giving guns and explosives to the players, and only allow a maximum of two players to interact with each other at a time. When the player is alone and another player can be seen far away,

the other player starts to become really interesting. To remove distraction of emotions, Chen removed the possible of chatting and seeing the non-diegetic player IDs, which could break the immersion. [5]

From a developer's perspective, user experience in games is a tool for a more player-centric development process to ensure that the designed experience is truly reflected in the player's mind. In the game team, everyone can be seen as a designer. Every choice for the game, even very small ones, can have minimal or large impacts on the player's experience. The infinite number of choices will easily detach the developers from the real impact of the choices made. By knowing the consequences of choices and how those affect the player's way of thinking and behaving, these will help the development team to iterate the design and justify design choices. This is the point where user-experience staff is needed. Throughout game development, UX will provide guidance when helping to design the game and user testing with players. [17]

The true meaning of UX is to realize how players experience the game. By that time, the development team will become blind to some aspects or find the game fun because they already know how it should work. If players do not understand the game, they will not find it good. By not knowing how players experience the game, it can make the development process even more confusing and challenging. Some important questions are "will players understand the rules?", "is the number of features enough or too many?" and "is the game balanced?". Not answering these questions—or even worse, answering them incorrectly—will have undesired negative consequences on how players experience the game. [17]

2.3 MOBA games

MOBA is an abbreviation for multiplayer online battle arena and it can be seen as a sub-genre of real-time strategy (RTS) games while inheriting some of its functionalities. While RTS focuses on macro-management, which entails commanding large amounts of units and constructing bases, MOBA focuses on micro-management, which is mastering a set of actions and their best use when controlling a small set of units. [18] Unlike in RTS games, there is no unit or building construction in MOBA games. Instead, the strategy is tied to individual character development and team cooperation. [19]

In most MOBA games, the player controls a single unit referred to as the hero. A hero has special skills and abilities that are used while in combat with other players. At the start of a match, every player chooses a hero to control from a large list of heroes with different skills and types. At the beginning of the game all heroes start at level one and they keep leveling up during the match. Common game modes in most MOBA games are played by two teams. In one common game mode, both teams have a base containing a structure. The team that destroys the base of the opposite team wins.

A general MOBA map consist of three lines that all contains three turret towers. The generic map is shown in **Figure 3**. Attacking to the enemy base happens through three lines and jungle areas between them. The middle line is a straight path from one base to another, while the other lines goes through the outline of the map. All three lines have

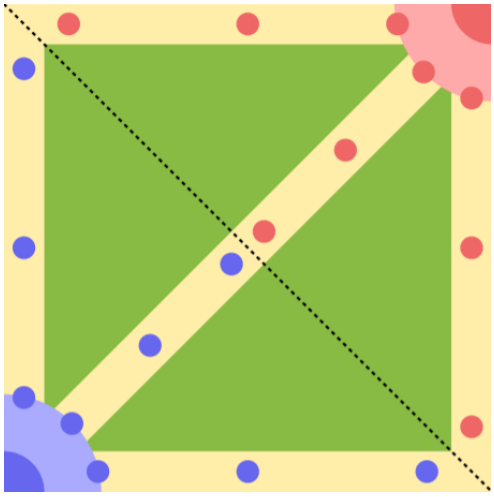

Figure 3. A generic map of most MOBA games. Blue and red circles indicate friendly and enemy towers respectively. Green area indicates the jungle areas. [18]

towers in them. Dividing the map evenly for both teams, the towers in the lines belong to the team that 'owns' the half of the map where the tower is located at. By destroying these towers, players can proceed towards the enemy base. The team that destroys the base of the opposite team wins.[18]

League of Legends and Dota 2 were the two most played MOBA games in November 2015 and they have similar features in terms of core gameplay [20-22]. Both of these games have a fantasy setting, 5-vs-5 gameplay, hero upgrading during the match, waves of minions that will help to attack the enemy, and similar map layout. The maps in both games care similar to the generic map in **Figure 3**. [21,22]

Recently, MOBA games have seen growth and become the majority of PC gaming. MOBA games constituted 30% of total play time in PC online gaming in November 2015. League of Legends itself amounted to 22.92%. [20] It was the most played game in the North-America and Europe while overtaking the MMORPG game World of Warcraft during 2011-2012 [23]. The second most played game in 2015 was Counter Strike: Global Offensive, which is a first-person shooter (FPS) game and had 6.88% of the total gaming time [20].

In November 2017, there were five big MOBA games on mobile phones. The global revenue distributed among those were the following:

- Arena of Valor – 71.4%
- Mobile Legends: Bang Bang – 21.2%
- Heroes Evolved – 4.0%
- Vainglory – 2.2%
- Heroes of Order & Chaos – 1.2% [24]

3 Mobile games

The game the author is making with his team is intended for mobile devices: i.e. phones and tablets. As the goal of the game is to make it work in portrait mode with different screen sizes and playable with one finger, we have to know the ergonomics of one-finger games and UIs, the kind of inputs that are to be used, and how to test the game. Furthermore, UIs of some existing MOBA mobile games will be analyzed.

3.1 Design

MDA framework is a supportive tool for game design and tuning. MDA stands for Mechanics, Dynamics and Aesthetics, which refers to the rules, the system, and the fun features of the game, respectively. Developers create the game in the MDA order, whereas players experience the game in reverse (i.e. ADM). The utility of MDA is that it helps to observe the impact of even small changes on other aspects and features of the game. [14] For the purposes of this exposition, the analysis will begin with aesthetics and end with the mechanics, as this is in line with the user-oriented approach.

Aesthetics refers to the game components such as sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission [14]. With reference to the MDA framework, the game to be developed, the aesthetic components are challenge, fellowship, expression, and fantasy. As it is a MOBA game and thus heavily focused on online gaming, the challenge will be the main role in the game. Players have to team up and find strategies to defeat the opponent team. Fellowship comes from the team play and social interaction. Players are meant to be able to create and join guilds and play online matches with friends. This creates a social interaction to the game. Expression and fantasy come from the core gameplay and meta game and by choosing their character and modifying its outfits and skills.

Dynamics are the system that create aesthetic experiences [14]. In the game to be developed, the various number of different heroes and different skills will create many possibilities how to play the game. The player can choose to be running in the frontline, supporting friendlies, or to use any other tactics they find useful or fun. Clear winning conditions and feedback of leading team will be given for not losing players' interest [14].

Mechanics are the numerous different actions, behaviors, and controls within the game context. These include the basic context and items in a game. For example, card games have shuffling, cards, and betting, whereas shooter games have guns, ammunition, and spawn points. [14] The game to be developed will contain a various number of different heroes with different and unique skills. Every skill has its own functionality, damage, cooldown time, and range.

While iterating and tuning the game, the aesthetics lets developers focus on what really is wanted and needed for the game. Dynamics describes the gameplay structure and it can be used to pin point areas that affect problems. Even though the goals of game design are defined usually by dynamics and aesthetics, developers can only directly influence mechanics. [14]

3.2 Functionality area of the thumb on touchscreen surface

The game the author is making with his team is intended for mobile devices: i.e. phones and tablets. As the goal of the game is to make it work in portrait mode with different screen sizes and playable with one finger (namely, thumb of hand holding device), we have to know the ergonomics of one-finger games and their UIs.

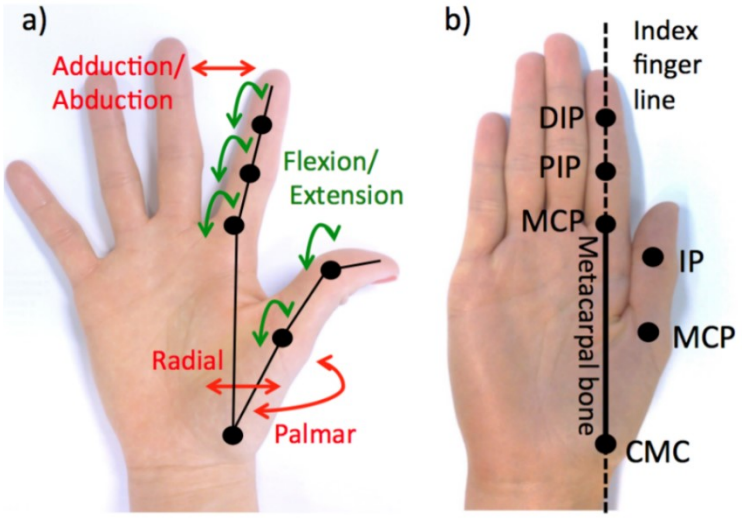

Figure 4. Terminology of joints. a) Joint movement adduction, abduction, extension and flexion. b) Joint names and index finger line which connects index finger and metacarpal bone. [25]

In Bergstrom-Lehtovirta and Oulasvirta's study, they modeled the functionality area of the thumb on mobile touchscreen surfaces [25]. Before taking a look at the modeling, some terminology used for the model of finger joints has to be covered. **Figure 4** shows the terminology used for joint movements and joint names and demonstrates the index finger line. The index finger is connected to the thumb with metacarpal bone and, therefore, the index finger has an effect on the thumb's movement. Next, we turn to the dependence between the index finger and the thumb. The distal interphalangeal (DIP) and proximal interphalangeal (PIP) joints in the index finder allow flexion of 110° and extension of 65° [25]. The metacarpophalangeal (MCP) joint allows for 85° flexion. The metacarpal bone connects index finger's MCP joint with thumb's carpometacarpal (CMC) joint. The CMC joint enables thumb's flexion and extension in rotatory planes. The CMC joint allows palmar abduction ranges of 45° and radial abduction of 60°. Thumb's MCP joint allows flexion of 55°. The interphalangeal (IP) joint allows flexion of 70°–80°. [25]

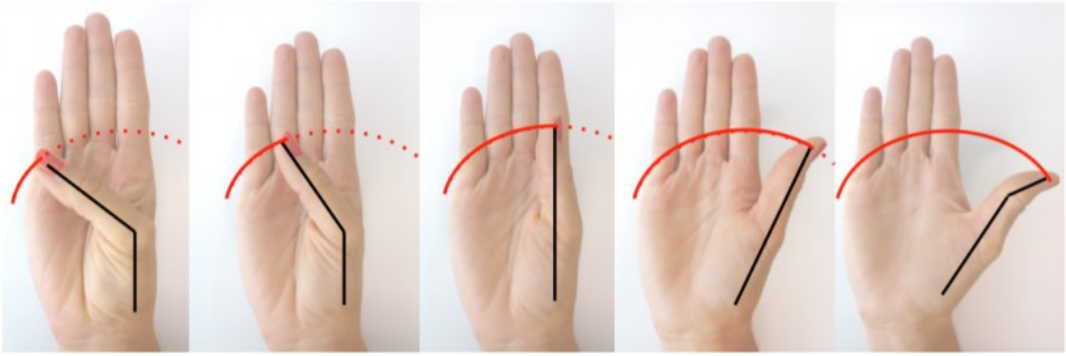

Figure 5. The motion of the thumb. [25]

For the following modeling, Bergstrom and Oulasvirta expected that the index finder is extended and adducted, in other words it is being straightened. When the hand is gripping the mobile device while keeping the index finger in a single position, the motion of the extended thumb on the surface defines the border of the functional area. In **Figure 5**, the motion of an extended thumb is illustrated. [25]

In the following, the point in the edge of the mobile device where the index finger line crosses the edge is called the origin. While gripping the mobile device, the distance of the index finger's tip from the origin has an effect for the thumb's functional area because the index finger is physically connected to the thumb's CMC joint and thereby affects considerably the pivot point of the thumb's motion. This distance is marked as d. Decreasing the value of d will affect the transform of the grip while giving thumb's CMC joint more space to move. When the fingers that support the device are in full length and straight, the thumb's CMC joint movement is heavily restricted, allowing only the MCP and IP joint to move. In contrast, when the supporting fingers are not in full length and straight, the thumb's CMC joint and the entire thumb can move freely, thus greatly increasing the function area on the device's touchscreen surface. The impact the distance

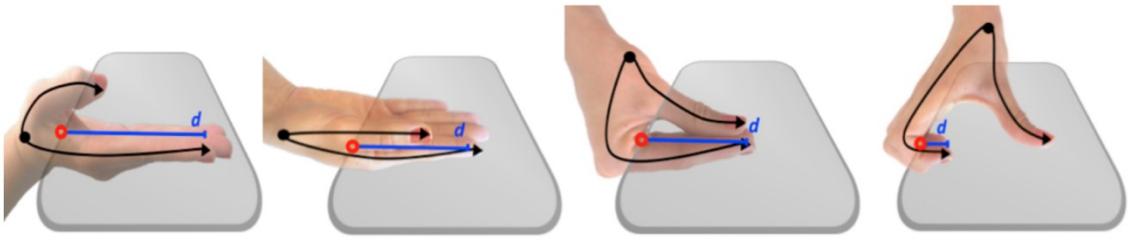

Figure 6. The images illustrate how the distance of d has an impact to the functionality area of the thumb on touchscreen surface. [25]

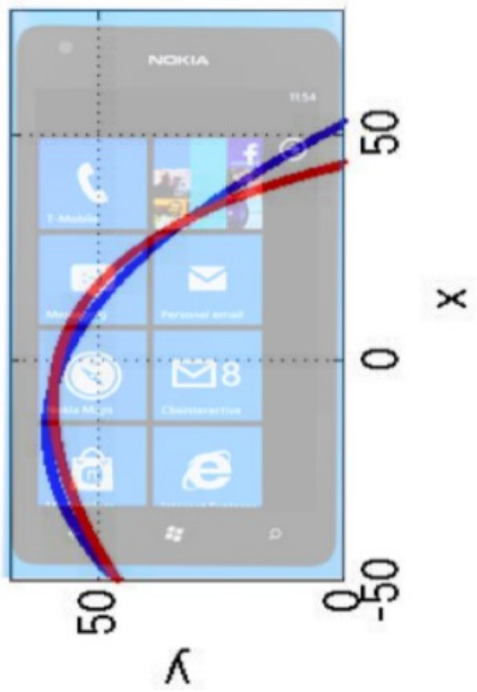

Figure 7. Functionality area of the thumb. Blue line illustrates α = 90° and red line illustrates α = 110°. [25]

d applies to the functionality are of the thumb on touchscreen surface is illustrated in **Figure 6**. [25]

Figure 7 illustrates the functionality area of the thumb. The origin is in the point where index finger line crosses the edge of the device. Angle α indicates the angle between index finger and the vertical edge of the device. If the angle is 90° then the blue line is used, and if the angle is 110° then the red line is used to show the functional area of the thumb. [25]

3.3 Inputs

Studying game inputs happens at three levels: micro, macro, and tactile. The micro level is about examining the individual inputs in the input device. The first measurable aspect is the number of buttons. As an example, an Xbox 360 controller has 15 buttons. [26] Inputs can be divided into two main categories: discrete and continuous. Continuous inputs send signals continuously whereas as discrete inputs send signals momentary. Keyboard keys, mouse buttons and controller buttons are discrete inputs. Continuous inputs are, for example, a mouse, joystick, and a steering wheel.

These controls can be categorized by the following aspects: type of motion, type of sensitivity, motion dimension, boundaries, direct vs. indirect control, position vs. rate control, and integral vs. separable dimensions. The type of motion divides inputs into two groups: linear measurements in two dimensions and rotation, for example, a mouse and a steering wheel, respectively. The type of sensitivity divides the inputs by the input measurement type. A mouse measures linear movement in two dimensions whereas joystick measures the force applied to it. [26,27] Movements can be measured in one to three dimensions. As previously mentioned, a mouse measures movement in two dimensions. Trigger buttons measure the movement in one dimension, while, for example, Wii remote controllers measure movement in three dimensions. A mouse has no physical boundaries of movement area, while thumbsticks in controllers have a round casing enclosing it and therefore bounding how far the stick can be moved. [26] A mouse is an indirect input because moving the mouse on a table reflects the movements on the screen. Touchscreens are direct inputs. Position vs. rate control divides the inputs by their way of handling the cursor, for example. A mouse changes the position of the cursor while joystick changes the velocity of the cursor. Integral vs. separable dimension focuses the inputs by the fact of effected dimensions. A mouse can control the cursor in two dimensions, therefore being an integral input. In a pair of knob kneels such as in an Etch-a-Sketch toy, both kneels controls only one dimension being hereby in the separable dimension class. [27]

On the macro level, the analysis focuses on the inputs as a whole input device and the potential of combining inputs. To understand the input space of an input device, the sensitivity of each button has to be known and how the layout and design of the device affects the sensitivity of the whole device. For example, a Nintendo Entertainment System (NES) controller has eight buttons. Four of the buttons are placed on the left side of the controller as a D-pad, two buttons in the middle and two buttons on the right side of the controller. Two middle buttons are not used during core gameplay and are hereby ignored. The D-pad has exclusives that reduces the sensitivity of the controller. These exclusives do not allow the up and down buttons or left and right buttons to be pressed simultaneously. However, the layout allows the use of multiple thumbs and therefore

increasing the sensitivity by increasing the number of possible inputs by combining individual inputs. For example, the player could press the two right most buttons at the same time or press one of the buttons on the right and press up and right on the D-pad at the same time. [26]

On the tactile level, the analysis focuses on the actual input device and how the inputs feel physically. The tactile of the input device has an impact to the feel of a gameplay experience. For example, a steering wheel may feel much better than a keyboard when playing racing games. Nevertheless, this cannot be measured and is mostly based on opinions. The input device has three important aspects: weight, materials, and input quality. Heavier and solid-feeling controllers are often seen as a better quality. In general, controllers are light in weight and this has a large impact on the feel of the controller. Similarly, to the factor of weight, the material choices of the controller also effects the feel of the controller but not the actual gameplay. The controller should be tight or loose, depending on the user's preferences, and quick-responding. To achieve these features, one important piece of the input device are its springs. The quality, construction, and the type of the springs has an effect on the input whether it is a general button or joystick. [26]

3.4 Playtesting

Tracy Fullerton highlights the importance of playtesting in her game design workshop book. She also mentions that designers do not always understand playtesting. There are multiple ways for testing a game with different perspectives and outcomes. Internal design review is when the designer and the developer team play the game and talk about the features in it. Quality assurance testing is having the quality assurance team go through the whole game trying to find bugs and flaws in the software. Focus group testing is interviewing players of the game, the possibility of them buying the game, and at what price. Tracking the players' mouse movements, eye movements, and navigation are considered to be usability testing. However, Fullerton focuses is on playtesting. [28]

Playtesting is having multiple players playing the game while the designer is collecting feedback. The overall goal is to gain useful feedback from the players to improve the overall experience of the game. Playtesting should start as fast as possible when there is something meaningful to test. Beginning the test phase when the game is in beta release is too late, and then it is hard to make large changes to the gameplay. Playtesting can be started by self-testing to ensure that the game works as intended. As the project progresses, the designer has to rely more on the outside testers to get an accurate understanding of how the game really works. The ideal in playtesting is that the designers should not need to tell anything about the game. The test players should have enough information that can be gleaned from the game itself. Friends and family are bad game testers after very first game tests because they are more likely to be too harsh or too unforgiving. Having players who do not know the designer or the game as game testers is a better idea for getting more honest feedback, because the testers will not gain or lose anything regardless what they say about the game. [28]

By getting the most up-to-date build and testing it with test players, the development team will gather good information of player experiences and behavior of the current game version. By repeating this week by week, the test sessions can be focused on specific areas. Players should be different at every test session to prevent players that already

know or assume some functionality from impacting the testing and to revisit the learnability of the game mechanics each time. [17]

While starting the playtesting session, it is important to remember not to tell anything about the game unless it is really needed, such as missing crucial information in the game. In playtesting, there has to be an investigator and an observer that must give testers access to the game, lead them through a useful play session, record what they do and say, and afterwards analyze their actions and responses. The basic guideline for testing is to have following phases: introduction, warm-up discussion, play session, discussion of game experience, and wrap-up. [28]

The introduction includes welcoming the player and telling the brief explanation of the playtesting process. In potential test cases when recording video or sound, approval has to be asked while telling that the recording is used only internally for analyzing the gameplay. Also, introductions to other persons in the room or behind a one-way glass should be done. In the warm-up discussion, some game habits can be asked from the testers like what kind of games they have played, what did they like of those, and what game they have purchased more recently. In the play session, it should be mentioned that the subject under investigation is not the player's skills but the game itself, and that the game is not ready yet as it is in development. The observer should either leave the tester alone and watch the gameplay behind a one-way glass or via video feed or stay in the room quietly behind the tester. The player should be asked to think loudly so the player's every action or thought would be understood and analyzed correctly. The length of the session should be approximately 15 minutes or otherwise the player may get tired. All the player's questions should be first answered with a question that provides room for further input: "What do you think what should be done here?" If the tester gets badly stuck, guidance can be provided and it should be written down where and why the tester needed help. This phase is followed by discussion of game experience where a one-to-one discussion is held about the game experience. Example questions can be the overall thoughts about the game and gameplay, what was the objective in the game, what they did not like, and what was confusing and why. While the project progresses, the questions should become more specific. In the wrap-up, the test session will be concluded, the testers will be thanked, and possible promo gifts given. [28]

Collecting and hearing feedback from players can be hard in some situations if negative feedback is not appropriately internalized. Sometimes it is just easier to respond and give excuses for negative parts and only focus on positive sides. Nevertheless, the playtest session was held also for learning what is not working in the game. From this perspective, the negative feedback cannot be skipped. Negative comments are better to be received immediately than when the game is published. The group testing has the concern of having one tester affect the opinions of others. One way of preventing this is to isolate testers from each other. If that is not possible, some ground rules should be made clear. The session is open for feedback, every comment should be honest and is appreciated. Everyone should respect on others' opinions and there are no wrong answers to any questions. [28]

It is important that all feedback and observations are written down. Trying to remember all the feedback and observations will lead one to remember only those aspects that were already known or guessed. It is good to write notes chronically with dates. The number

of questions asked in a row should be limited to or less than 20, otherwise testers may get exhausted and not answer accurately. [28]

One way of collecting quantitative data is to count how many clicks it took for the testers to do some tasks or how long it took to get to a certain point in the game. Also, the testers can be asked to rate some functionalities or asked how well these worked from 1 to 10. Before collecting data, it is good practice to think about the assumptions and purposes of the testing aspect. Then the test can be structured in a way that it will affirm or deny the hypothesis in its conclusion. The qualitative approach is to ask how the testers felt some new given features. Data can also be collected automatically by the game code, which saves data of how the players played the game. It is important to realize that new players can act very differently to players who have played the game more. Therefore, the data and the feedback can vary by what is tested and with who. Combining different data analysis in playtesting will give the best overall results. [28]

The effectiveness play session can be improved by creating test control situations. To test the end of the game, the testers do not have to start the game from the beginning and play through the whole game. Instead, testers can start the game at or near the end to ensure that the player will reach the end. Other possible testable situations are testing the basic functionality without concern for game balance, a specific level, a special situation in the game, new features or random event during the gameplay. The purpose of test control situations is to make faster tests on a specific area and to test situations that would be rare or almost impossible during gameplay. Cheat codes are one good way of applying test control situations by, for example, allowing the player to have infinite resources. [28]

3.5 MOBA mobile games

When writing the word ‘MOBA’ to Google Play or App Store, a long list of MOBA mobile games will be listed. On January 7, 2018, the first five listed fully released MOBA games in the Finnish Google Play were:

1. Arena of Valor: 5v5 Arena Game
2. Mobile Legends: Bang Bang
3. Vainglory
4. Heroes Evolved
5. Heroes Arena

On the same day, the first five listed fully released MOBA games in the Finnish App Store were:

1. Mobile Legends: Bang Bang
2. Arena of Valor
3. Vainglory
4. Ace of Arenas - The Mobile MOBA Mastered
5. Heroes of Order & Chaos

Arena of Valor: 5v5 Arena Game and Arena of Valor are the same game even though they are named differently in different stores. Next, we are going to look some UI aspects in the previous games. In addition to these games, also the UI of Brawl Stars will be analyzed.

The common thing among these games excluding Brawl Starts is the fact that they are all played in landscape mode [7,29-35]. The following UI analyses are described as the author has understood them after playing couple of online matches against other players.

3.5.1 Arena of Valor

Arena of Valor is played in landscape mode and it has two-fingers gameplay. One of the gameplay modes is the same as in Dota 2 and League of Legends mentioned in chapter 2.3. In this game mode players have to defeat enemy base by attacking through three different lines full of enemy towers. [7]

Figure 8. The UI of Arena of Valor. Player is targeting with the 3rd skill (right most skill button). [7]

The UI of the game is shown in **Figure 8**. In the game, a hero has default attack, three skills, and three other abilities that give the hero, for instance, health, faster movement or teleports them back to the own base. These are all implemented as buttons on the right bottom corner of the screen. Skills and the default attack button can be used to attack automatically without targeting by pressing and releasing a skill or the default attack button. If a skill has manual aiming, it can be started by pressing and dragging the skill button. Skills show the width of the spell and its maximum range by projecting, for example, a circle and a rectangular sector on the ground as can be seen in **Figure 8**. By pressing the default attack button, the hero will target an enemy and will keep hitting until the enemy dies. After that, the hero will automatically select a new target, move towards it, and again hit the enemy until it is dead. Automatic attack can be cancelled by moving or using skills or some abilities. Other ability buttons are located at the bottom of the screen next to the skills. The effect of these buttons is explained with texts and accompanying images. [7]

3.5.2 Mobile Legends: Bang Bang

The UI of Mobile Legends: Bang Bang is very similar to the UI of Arena of Valor. It is played in landscape mode with two fingers and the UI layout is almost identical to Arena

of Valor. Screenshot of the game is shown in **Figure 9**. The main differences in the UIs of the two games are the layouts of the score information, team information, money location, and shop location. [7,31]

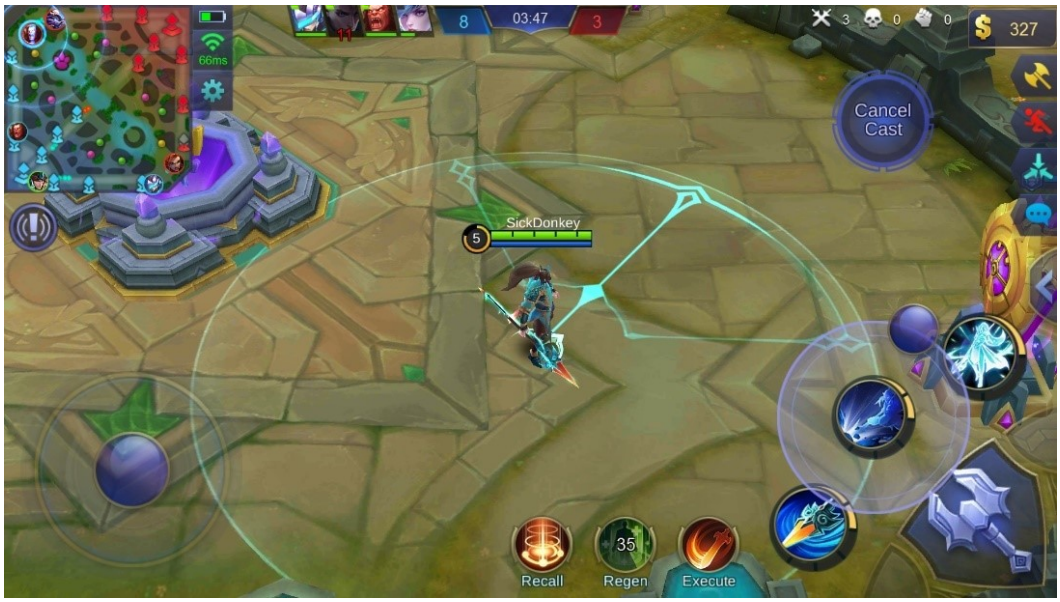

Figure 9. The UI of Mobile Legends: Bang Bang. The player is aiming with the second (middle) skill. [31]

3.5.3 Vainglory

Vainglory is a landscape mode game and its UI differ significantly from the UI of Arena of Valor [7,33]. The UI is shown in **Figure 10**. First, the game can be played with only one finger but the game highly recommends to play with two fingers. Two-finger gameplay allows an easier and faster way to interact with the game. Vainglory is played mostly by tapping. Moving a hero happens in two different way. Tapping the screen will command

Figure 10. The UI of Vainglory. Player has activated a skill which needs a target by tapping on an enemy. [33]

the hero to move to the tapped position. Another way is to hold a finger on the screen. Then the hero will move continuously towards the finger's direction. A hero has three skills and also different abilities that can be purchased during a match. Attacking enemies happens also by tapping. Tapping on an enemy marks that enemy as the target and the hero will start to use its default attack to kill it. After the marked enemy is killed, the hero will automatically pick a new target if there are possible targets close enough. If the enemy is marked but too far to be attacked, the hero will automatically move next to it. [33]

Skill are located at the bottom middle of the screen. Activating these skills happens two different ways depending of the skill. Some skills activate immediately when the skill button is pressed. The other skills only show a maximum targeting range after pressing the skill. After that a target for the skill has to be marked by tapping on an enemy or a structure. [33]

The game mechanisms are easy to learn. However, while moving towards a desired direction by holding a finger on the screen, some parts of the screen where the player is running are blocked [33].

3.5.4 Heroes Evolved

The UI in Heroes Evolved has many similarities with the UIs of Arena of Valor and Mobile Legends: Bang Bang [7,31,32]. The UI of the game can be seen in the **Figure 11**.

Heroes Evolved is a landscape game with two-finger gameplay. Moving happens with a virtual joystick that is located at the bottom left corner of the screen. On the right bottom of the screen are the default skill button and three special skills next to the button in the shape of an arc. While the player is using skills, a range indicator is shown to the player to indicate maximum range for attack. In attacks that damage enemies in a sector of a circle, a circular sector is shown in addition to a range indicator. Skills that do damage on a line show a path indicator to visualize the aiming direction as can be seen in **Figure 11**.

Figure 11. The UI of Heroes Evolved. Player is aiming with the 2nd skill. [32]

While the player is aiming with the skills, new joystick is created at top of the skill button. While the player moves the finger to choose aiming direction, the skill button moves inside the new joystick. By pressing the default skill button while being close enough to enemies, the hero character starts to kill enemies automatically. When the target enemy is dead, a new target is picked automatically if there are new potential targets close enough. Target enemies are marked with a red circle underneath them. [32]

3.5.5 Heroes Arena

Figure 12. The UI of Heroes Arena. Player is aiming with 4th skill. [35]

Heroes Arena is a two-finger landscape game. Its UI is similar to most of the previously described games. Unlike those games, Heroes Arena has four skills in addition to the default attack and it does not have any assisting text descriptions in its bottom ability buttons. The UI of the game can be seen in **Figure 12**. [35]

When an enemy hero is in the player's view area, a new button appears next to the skill buttons with an image of the enemy hero. By tapping the button, the corresponding enemy player is marked as a target. When the enemy hero is marked as a target and when the player starts to use its default attack, the player starts to attack the target instead of other possible enemies next to the player. [35]

3.5.6 Ace of Arenas

The UI in Ace of Arenas differs from the previously mentioned games. The UI is shown in **Figure 13**. A hero has a default attack, three skills, and three abilities. The default attack and three skills are located on the bottom right corner of the screen. The ability buttons, which are colored as grey, are located under the minimap. The UI does not show the movement joystick while the player is not using it. The movement joystick differs from the other previous games also by its functionality. Whenever the player tries to drag the finger outside of the movement joystick, the joystick starts to move towards the finger keeping the finger inside the joystick. [29]

Figure 13. The UI of Ace of Arenas. Two enemy heroes are in the game view of the player and there are two new buttons pointing towards the enemies. [29]

Choosing a target for the player's attacks happens by tapping on an enemy. When a player from the opposing team is fully in the players game view, a new button with an image of the enemy appears next to the skill buttons. Also, an arrow and dotted line are shown from the button towards the enemy, as can be seen also in the **Figure 13**. An enemy player can be chosen to be the target also by pressing the new button. Pressing and holding a finger on the default attack button creates a new UI element that covers the skill buttons. This new UI element contains five different default commands for other players. A desired command can be selected by dragging a finger towards it. These commands include, for example, telling teammates to attack, defend, or retreat. [29]

The default attack button has three functionalities during combat. By swiping up or down on top of the default attack, a targeting preference is set to champion (enemy heroes) or mob (minions), respectively. Pressing the default attack will start to attack on an already selected target, or if there is no target at the moment, it selects a new target by the preference and starts to attack the new target. [29]

3.5.7 Heroes of Order & Chaos

The UI and game mechanics of Heroes of Order & Chaos stands out from the aforementioned games. The UI of the game can be seen in **Figure 14**. The movement joystick is located at the bottom left corner of the screen, it is not visible while it is not being used, and its size is the same as in the ability buttons. Alternatively, the player can move the hero by tapping on the ground and then the hero walks towards a given position. By moving the hero by tapping the ground, the camera is not moved. If the player wants to move the camera, the player has to start dragging a finger outside of the movement

Figure 14. The UI of Heroes of Order & Chaos. [30]

joystick's working area, which is approximately at the bottom left quarter of the screen. Starting to drag finger inside movement joystick area will start movement by joystick. [30]

A hero has a default attack, four skills, and three abilities. The default attack button is located close to the right bottom corner of the screen, and in **Figure 14**, it has an image of a sword. The button appeared quite hidden to the author due to its small size and grey transparent color. Skill buttons are located on the right edge of the screen. The ability buttons are located at the bottom of the screen and these abilities can, for example, heal or teleport the hero back to the own base. Choosing a target enemy takes place by tapping on an enemy or by pressing the 'change target' button, which is located at the right bottom corner of the screen. [30]

From all previous MOBA games, the author had the most difficulties in understanding the UI in Heroes of Order & Chaos.

3.5.8 Brawl Stars

Brawl Stars is a portrait MOBA game made by Supercell and it was been soft launched to App Store in Canada, Finland, Sweden and Norway at the time of writing this thesis [34,36]. It is mainly a 3-vs-3 game with multiple game modes from collecting gems to defending a safe from the other team. Despite the portrait mode, the game is played with two fingers. One finger uses the movement joystick on the bottom left corner of the screen and the other finger is used for aiming and shooting. The UI of the game is shown in **Figure 15**. [34]

Aiming starts when player presses anywhere on the screen. There are three kinds of aiming in the game. The first type is a circular sector and it indicates how far the player can shoot and how wide the shooting affects. This kind of aiming is shown in the left image of **Figure 15**. The second one is line aiming. Just like in the circular sector, the

Figure 15. The UI of Brawl Stars. On the right image, the player is ready to activate a special skill (yellow button). [34]

line shows how far the player can shoot and how wide the affected area will be. The circular sector and line start from the player and are turned towards the aiming finger's position. The third aiming type is arc. In this aiming type, the player chooses the target position for the skill by the aiming finger. If the player tries to set a target position that is too far for the player's character, the furthest allowed point in the direction of the finder will be marked as the target position. There is no indicator for showing the player from how far the character can choose the target position. Arc aiming is shown in the right image of **Figure 15**. By releasing the finger, the player starts its default attack in the given direction or position depending on the attack. [34]

After receiving enough experience points, the player can activate special skill for single attack by pressing the special skill button on the bottom right corner on the screen. The special skill can be deactivated for later use by pressing the same button again. Aiming and shooting rules are the same for the special skill, but the aiming type may differ from the default skill. When shooting with the default skill, the player uses energy. If there is not enough energy for the skill, the player has to wait until the skill can be used again. The special skill does not use the energy. The current amount of energy is shown under the health bar. [34]

While running and trying to shoot towards the bottom left corner of the screen, on small devices the moving finger and aiming finger easily collide making the running and shooting cumbersome. [34]

After an update in March 2018, the orientation of Brawl Stars was changed from portrait to landscape mode. The new UI can be seen in **Figure 16**. [37] It seems that the developer team came to a point where they found the game working better in landscape mode. This change may allow for easier aiming or at least a clearer UI, because players can now aim with a joystick that is located at the right side of the screen [37]. This will decrease the amount of UI blocked by the aiming finger. Also, automatic targeting and shooting was added to the game. This happens by tapping the aiming joystick. Tapping the aiming button will automatically find the nearest enemy. If the nearest enemy is close enough to be shot, the shot is fired. The ultimate skill is aimed from its own button and it can be shot automatically by tapping the button after the skill is ready to be used. [37]

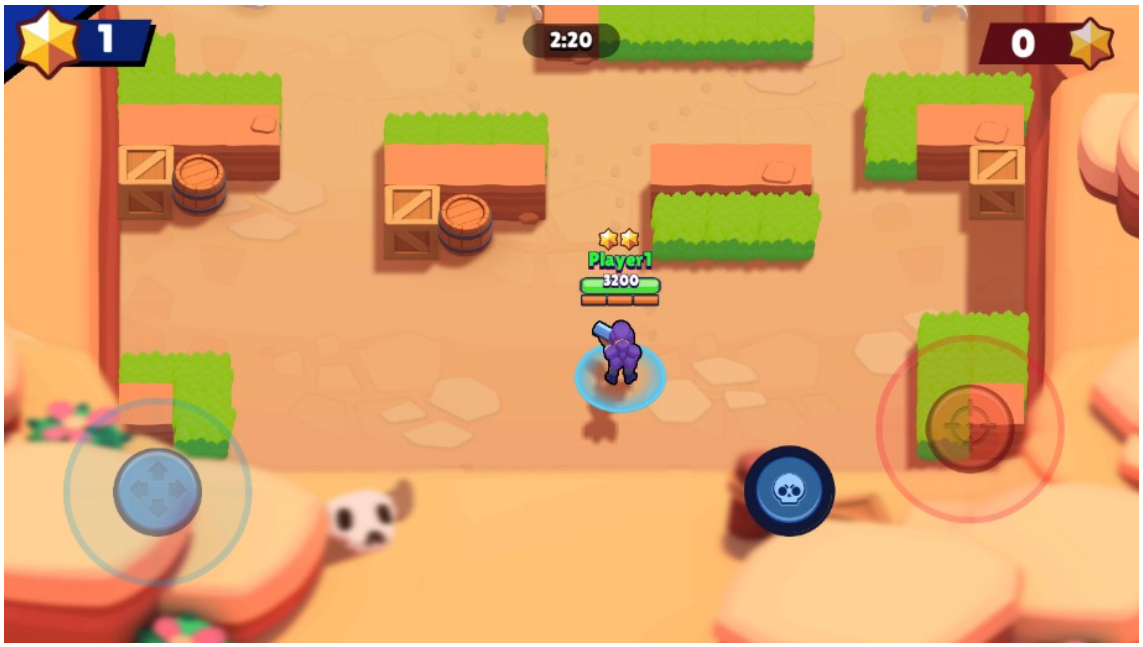

Figure 16. Brawl Stars on landscape mode after an update on March 2018. [37]

4 Developing a MOBA mobile game

The aim of this master's thesis was to optimize a MOBA mobile game in a way that new players could learn how to play the game immediately. Players should be able to play without instructions in the beginning or middle of gameplay and they should not need any help in performing actions or using the user interface during the gameplay.

The main focus in the project was to design and develop a UI for gameplay of MOBA mobile game. Portrait mode was the further aim for the game to ensure a more casual game experience for players, as portrait mode allows easier one-handed gameplay than landscape mode. More specifically, the aim was that the player plays the game with a thumb of the hand holding the mobile device.

The target of the UI design was to provide easy-to-understand and easy-to-use gameplay interactions. Many alternative UIs were to be made and tested with test users. The first user tests were to be made by playing against bots. In these conditions players were able to try and learn the game mechanics without caring about other players. Later player versus player tests were made to get results how players learn the game mechanics during combats and did they enjoy the game.

In addition to the UI, some aspects of UX were analyzed to get a better understanding of whether the game experience is enjoyable or not. There is a possibility that the UI for the game is not successful in portrait mode, in which case the game will be created in landscape mode. The first possible problem areas identified in portrait mode are how the game works with different screen sizes and whether the fingers and other buttons cover too much of the gameplay area.

The MOBA mobile game was made with Unity3D engine.

This chapter covers the whole project including design, iteration process, analysis, and end results of designing a gameplay UI for a MOBA mobile game.

4.1 The beginning of the project

At the starting point of this project, the author was given a prototype of the game with limited actions and UI. This UI is shown in the left most image in **Figure 17**. Players were able to move the hero by virtual joystick in the direction of a player's input, use melee attack with the top right skill button, and throw a ball in the player's chosen direction by using the top left skill button as a joystick. Furthermore, the command to throw the ball was given after releasing the finger that initiated the aiming. Moving and throwing directions were given by first touching the below half of the screen or the throwing skill button, respectively, and then moving the finger in the desired direction. **Figure 17** shows also the UI for the movement joystick in the rightmost image. Also, potential ideas for future actions and UI design was given at the beginning of the project.

These future ideas were a required number of skill buttons, which was four, cooldown meters for abilities and other features that are commonly used in games (especially in MOBA games), such as health bars and a minimap.

Figure 17. The starting point of the UI for the author (left picture) and UI after first additions (middle picture). Right picture illustrates the UI of the movement joystick. After the starting point version, the area for movement joystick covered the whole screen.

4.2 Building the core of the gameplay UI

The first addition to the gameplay UI made by the author was to increase the number of skill buttons from one to three. The buttons where placed in rectangular shape as can be seen in the middle image of **Figure 17**. At this stage, three different types of skills were designed: line, high arc, and cone.

Line skills were used in cases where the player shoots a projectile in a given direction, just like in the ball throwing ability described previously. A player could aim into a desired direction and then activate the skill by releasing the pressing finger. The maximum shooting range was indicated by a circle UI indicator. Projectiles would not fly further than the circle indicated. The UI indicators of aiming with the line skill is illustrated in **Figure 18**.

With high arc skill, the player could mark the area where the player would throw spells. The spell would be thrown along a negative parabola. When the spell hits its destination, a predefined functionality would happen inside its effective radius, for example, healing all allies or damaging all enemies within the area. To activate this skill, that player had to press the corresponding skill button like in the line skill. After activating the skill, the player could drag the finger along the surface of the mobile device. The center of the affected area was projected on the ground where the player's finger was located. The high arc UI indicator is illustrated in **Figure 18**.

The cone skill also was activated by pressing and holding a finger on a corresponding skill button. If the player did not move the finger after activating the skill, the nearest enemy within a maximum shooting range was chosen automatically as the enemy target. The maximum shooting range was marked with a circle indicator. If the player moved the

Figure 18. UI indicators while aiming manually with different skills. Left image shows the aiming indicator for line skill. Middle image shows the aiming indicator for high arc skill. The right image shows the aiming indicator for the cone skill.

finger beyond a small threshold for a movement capture, the manual aiming was started. The player was now able to start aiming by moving the pressing finger towards the desired direction from the pressing point of the skill button. The hero character started to aim with a predefined angle in the given direction. While aiming manually, the maximum range was marked by a circular sector. The middle of the circle (thus the sharp edge of the circular sector) was centered by the player's position. The length of the circular sector indicator was the same as the maximum range. Also, the angle of the sector was predefined for every individual cone skill. The UI indicator indicated the size of the area where the enemies could be damaged. All enemies within the targeting area were selected as targets. If no enemies were located inside the targeting area, the skill could not be used. Furthermore, even though the cone attack had a specific target in automatic targeting, it hit all enemies within the cone area. The aiming indicator when manually aiming with cone skill is shown in **Figure 18**.

After starting to aim with any skill automatically or manually, two cancel buttons appeared on the screen. Cancel buttons were grey circles with a cross. One cancel button was added to both sides on the screen as can be seen in **Figure 18**. If the player dragged the aiming finger on top of one of these cancel buttons and released the finger, the skill was cancelled and ready to be used again.

At this moment, the game had three different working skills and a melee attack. Melee attack was used by clicking the melee button. Any enemies directly in front of the player would have been hit.

The second addition was to enable automatic aiming for the line and high arc skills to provide easier and faster attacking like with the cone skill. The automatic attacking happened by pressing the corresponding skill button without dragging the finger. The nearest enemy within a defined maximum range was marked as a target enemy. By

releasing the finger, the line skill shot a ball towards the enemy's current position and the high arc created a damaging area centering on the enemy's current position. Automatic targeting worked as long as the player did not start to use the button as a joystick by dragging it. After dragging the aiming was changed from automatic to manual aiming.

Target enemies (or friendlies) were also visually marked for the player. This target marker was a red spinning circle projected on the ground with its center point at the target player's coordinates. Target enemies were shown to the player when the player was aiming automatically or manually and when there was a target within maximum range, and in the case of the cone, at its maximum angle. All potential target players inside the cone area were marked as targets whether the player was using automatic aiming or not. The UI indicator for the target marker is shown in **Figure 19**. Furthermore, the projectiles in the line aiming were homing towards the target player if there existed one.

Figure 19. On the left image, the player is aiming with a line skill and the player has a target. On the right image, the player has two targets with the cone skill.

Using the high arc skill by dragging the finger along the surface of the mobile device was seen as a bad functionality. As seen in the chapter 3.2, not all areas of the touchscreen can be easily reached while playing with the thumb of the hand holding the mobile device. Also, with larger screen sizes such as tablets, players may have to drag the finger long distances and it was not seen as convenient or user friendly. As a result of this reasoning, a new way of aiming with high arc skill was created. The UI for this can be seen in **Figure 20**. After the player activates the high arc skill, a new black joystick that is a bit larger than the pressed button will appear behind the button. The player can now move the finger inside the joystick area and a new small green marker appears and follows the finger. If the finger moves outside of the joystick area, the small marker stops at the edge of the joystick. The small marker is mapped into the play area. The center of the joystick is

mapped to the player's position. The edges of the joystick area are mapped to the edges of the predefined maximum shooting range that is also shown to the player as a circle. By this functionality, the player was able to aim faster with smaller finger movements.

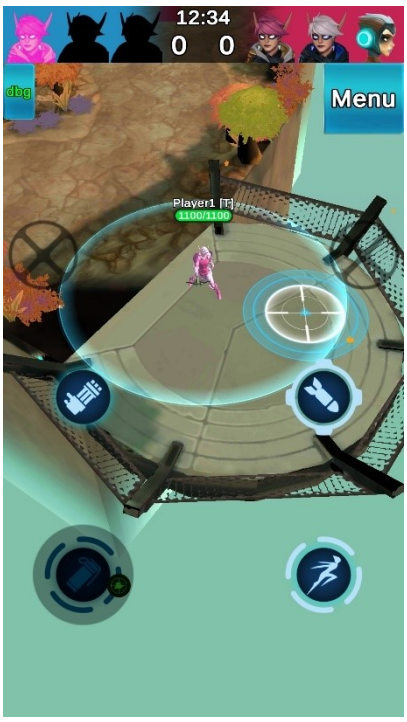

Figure 20. New way of aiming with the high arc skill.

In the next step, four different heroes were added to the game with different abilities and skills. This information was read from spreadsheets and the skill buttons were defined by these values. The assumption of having all different skill types was no longer holding. The skills were based on the chosen hero. Furthermore, the melee button was removed and replaced with a skill button changing the total number of skill buttons to four. Also, a new skill type, instant skill, was added to the game. Instant skill worked by pressing the corresponding skill button and releasing the finger. After releasing the finger, the skill was activated. This skill type did not have any kind of aiming functionality. The instant skill could have affected the player by, for example, giving the player faster movement or a multiplier for dealt damage, or it could have affected all friendlies or enemies within a predefined maximum range. If the skill had a predefined maximum range for healing allies or damaging enemies, this maximum range circle was shown to the player. The target players inside the maximum range were marked with the target markers. If the instant skill had only effects on the player itself, the player would have been marked with the target marker.

After any skill was used, it was put in cooldown mode for a predefined time. While a skill was in cooldown mode, the skill could not be used again until the skill was regenerated. The cooldown was marked on top of the ability button by a black circle that started to disappear clockwise.

4.3 Input analyzes

We will analyze the inputs of the mobile game in three categories using the input device analysis rules of Steve Swink: namely, micro, macro, and tactile levels. On the micro level we analyze all inputs of an individual. [26] As the input device is the player's mobile device, there is the one physical input (the touchscreen) and multiple virtual buttons on the screen. In the designing and prototyping phase, there were six buttons that can be seen in **Figure 21**.

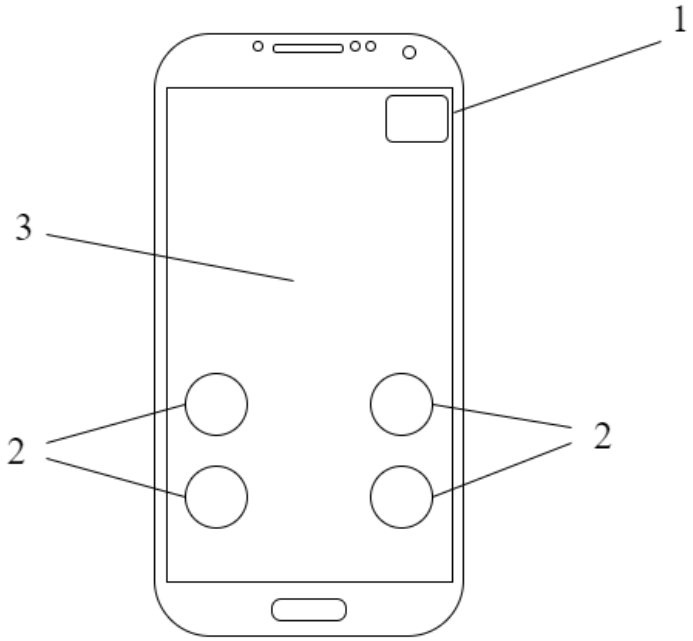

Figure 21. The first prototype design of the input locations and types.

The following button numbers refer to the buttons in the **Figure 21**. Button 1 is used for opening a menu. This is a standard button and it operates only if the player has pressed and released the button. The menu button will not be covered any further. All four buttons marked with the number 2 are skill buttons. Depending on the hero character and the skill types of the character, the buttons can have three different functionalities based on the input perspective.

In functionality 1, when the button is pressed and released, execution for a certain skill ability is sent. The instant skill has only this feature. The line, cone, and high arc skills also have this functionality as the aiming is automatic. In functionality 2, when a player presses a skill button and moves the finger in a direction of an angle α from the pressed point, the player's character turns in the same angle direction α on the world coordinates. When the button is released, the character attacks in a defined way in the given direction. The line and cone skills use this functionality. Even though the skills worked differently, they worked the same way from the input perspective. In functionality 3, when a player presses a skill button and moves the finger for a distance d of an angle α from the pressed point, a targeting area occurs in the game for a mapped distance d' for the same angle direction α on world coordinates where the player's position acts as the origin. If d = 0 then d' = 0. If d has its maximum value, also d' has its maximum value. These two last

mentioned button functionalities measure movement linearly in two-dimensional space on the touchscreen surface. They also have the same boundary, which is the touchscreen edge.

Finally, button 3 includes the whole touchscreen, excluding the areas of the other previously mentioned buttons, and works as a touchpad for the player's movement. When the player holds a finger in a point in the reserved area of the button 3, two new UI indicators occur: one for indicating the area where the player just pressed the button and the second for indicating where the player's finger is at the moment. The player's movement direction is the same as with the two UI indicators when the first indicator is the origin and second is a given destination. The distance of the two indicators has no effect on the player's movement direction (or speed), but only the coordinates of the second indicator when the first indicator acts as an origin. The button input measures movement linearly in two-dimensional coordinates on top of the touchscreen surface and the only boundaries are the edges of the touchscreen.

On the macro level, we focus on the inputs as a whole [26]. At this point, it is good to recall that the team's goal with the portrait UI is to make it work during gameplay in a way that the player can play the game only with the thumb of the hand holding the mobile device. To prevent the possible advantages of playing with two fingers, only one action can be executed at any given time. This means that the player cannot use two different attacks at the same time or attack and move at the same time unless the movement is automatic and provided by the attack. Furthermore, this aspect will have a strong effect on the final UI design.

On the tactile level, we analyze not only the input as a whole but also the device itself [26]. Since the team's game is a mobile game, the whole mobile device is the input controller for the team's game. The larger the touchscreen is, the larger and better the game screen is for allowing more precise actions. However, large touchscreens require large mobile devices that lead to heavier devices. Large and heavy devices like tablets are most likely placed onto a table or equivalent or held by the other hand while playing. While playing with a larger phone with a single hand, the holding fingers behind the device has to be straighter to compensate for the width and the weight increments. However, as we saw in chapter 3.2 when the fingers are more straight the thumb's CMC joint and the entire thumb has a more restricted movement, it allows for a smaller functionality area of the touchscreen surface. In conclusion, all mobile devices with different width, height, screen size, thickness, or weight have slightly different gaming feels in terms of the input systems and the thumb's functionality areas. Therefore, the design of the final UI has to satisfy all the different mobile devices.

4.4 Iterating the UI

After having the main core gameplay UI implemented, the iteration phase started. To allow easier one-handed gameplay, the buttons in the UI had to be organized again. Three alternative ideas were created in addition to the original one. All different layouts are shown in **Figure 22**.

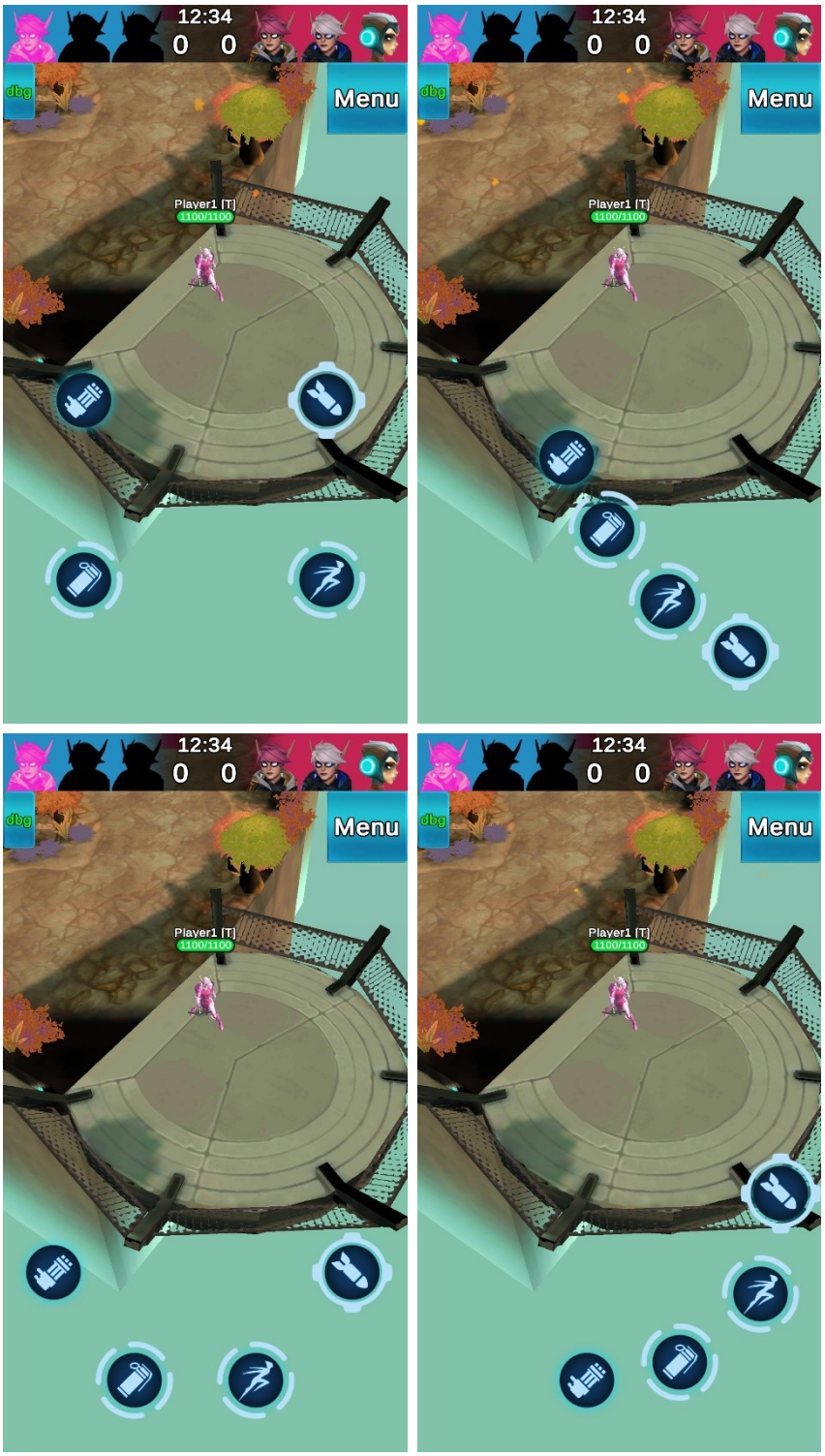

Figure 22. Alternative UI layouts. Names used to refer to these different layouts were: top left ‘square’, top right ‘left arc’, bottom left ‘trapezoid’ and bottom right ‘right arc’.

The main problem in the original ‘square’ layout was having the buttons largely separated from each other and the buttons were hard to reach while playing with a thumb. ‘Trapezoid’ layout brought the buttons a little bit closer to each other and allowed the player to use the gap between the buttons as the main area for movement. This made the

act of controlling feel more natural. Nevertheless, having the leftmost button on the opposite side compared to the rightmost button would create problems with larger screens while trying to play the game ergonomically with the thumb. The ‘left arc’ layout brought the buttons even more closer together, leaving some space for the movement on the right side of the arc. Even though the buttons were located in the functionality area of the thumb for right-handed gameplay, the buttons blocked some important areas in the below half of the screen. The ‘right arc’ fixed this problem by flipping the arc. This UI version had the easiest layout for switching between different spells and movement while avoiding blocking important game view areas.

At this time the team of this game project consisted of seven persons: the project leader, producer, game designer, artist, server programmer, and two client programmers. Later the group size grew to around ten persons. As game developers and designers, the team played the game multiple times alone locally, trying new features and ensuring that everything worked as intended. The team also played the game online against each other as teams with maximum sizes ranging between 2-5 players to test user experience of movement, skills, and the gameplay itself. During the play sessions, it was realized that the entire team were using two fingers more or less while playing. This may be due to the fact that the team were used to playing MOBA mobile games with two fingers: one for moving and one for using skills. This also occurred with the team’s game. At this point, the game did not allow the use of movement and skills at the same time, although using two fingers made switching between moving and shooting faster. Furthermore, it was

Figure 23. On the left image the enemy is marked to be the target for the shot with the help of aim assist. On the right image the angle between aiming line and direction to the enemy is too large and therefore the enemy is not marked.

realized that one reason for wanting to play with two fingers was the fast-paced tempo of the game; in other words, the faster the player made the shot, the faster the player could start moving again. This had a negative impact on aiming skills, especially when it was already a bit hard to aim and the game felt rushed.

To provide easier manual aiming with the line aiming skill, the nearest enemy to the aiming line itself was marked as the target. The calculation was made by calculating the angle between aiming direction and possible target position from the player. If no possible target was inside a predefined maximum angle, then no target was selected as it would not be natural behavior to automatically target enemies that are in, say, the opposite direction of the aiming direction. After releasing the button, the player would 1) shoot in the aiming direction if no target was marked or 2) turn towards the marked target and shoot towards it. Illustrations of this aiming helper are shown in **Figure 23**.

Changing completely the manual aiming in the line skill to half manual and half automatic had a huge effect on gameplay. Previously, the manual aiming was not seen as positive for two reasons. First, using the manual aiming is slower than using automatic aiming. With automatic aiming, the player can press and release the ability button and then the player has a large probability of hitting the target. However, when starting manual aiming, the player has to start dragging the finger and therefore has at least a small delay when compared to automatic aiming. The second problem was more troublesome. While shooting manually, the player had the full controls with the shooting direction but most often the player missed the hit. By adding the functionality of picking the closest enemy as the target player, it helped with this second problem. Now players also got hits with manual aiming while having the ability to choose the target enemy by itself. Nevertheless, the automatic shooting still had the advantage in terms of speed and this can be crucial in the gameplay where only moving or shooting is allowed.

A new feature was added to the game to slow down the pace of the game and to give more time for aiming. This feature was called *casting time*. After the player had released the finger to start an attack, a casting time was initiated. After the casting time was finished, the skill was activated. The casting time was informed to the player in the UI as seen in **Figure 24**. The casting time counter was shown as a large circle—centered by the player's position—which was made of small white circles. One by one, these white circles turned to blue in clockwise order. After the last circle was colored blue, the casting time was done and the UI indicator for it was hidden. For testing purposes, there were two different variants of the casting. In the first variant, the player was not able to aim, start a new spell, or move while casting. In the other variant, the player was not able to aim or start a new spell but was able to stop the casting by moving. In this situation the attack was not started and the skill was not put into cooldown mode.

Introducing casting time had negative and positive results. Before the casting time was added to the gameplay, the game felt occasionally disrupted when the player had to stop to start shooting. As a negative result of casting time, this disruption was longer and therefore worse. After starting to attack, the player was stuck before the spell was shot. However, casting time slowed down the game pace and mitigated the hurried feeling of

Figure 24. Casting time UI indicators. On the left image, the casting time is almost half way done. On the right image, the casting time is about to finish.

switching between moving and shooting states. Before casting times, two fingers were used most often while playing to let players switch the state faster. Another positive result was the fact that it improved aiming. By forcing the players to stop while casting a spell, players tried to make every shot count by aiming better.

With previous changes the game did not feel like it was progressing naturally or like a continuous flow but became rather fragmented. The gameplay could be described to follow these stages: move, aim, start casting, hope that the target will not move away from your aiming direction, shoot and probably miss, move, avoid enemy attacks, move, aim, etc. To prevent the number of disruptions in the game, automatic attacks were added to the game. For every hero, one of the skills was changed to attacking automatically. If an enemy player was located in front of the player within a maximum targeting range and inside the player's targeting angle of 90°, the attack was automatically activated towards the enemy if the automatic skill was not in cooldown mode. Usage of automatic attack is illustrated in **Figure 25**. Automatic attack was put in cooldown after it was used or after using or cancelling any other skills. It also could not be used while the player was targeting or casting with other spells. The maximum range of automatic skills were in general much smaller than other manually used skills to allow for using automatic attack only in close range situations. The damage made by these attacks were also reduced to make them be only good secondary attacks, not primary attacks. Automatic attack also had one large difference compared to other spells. It did not put the player into casting mode before shooting the spell. This allowed the player to move and shoot at the same time. But with the other skills, the player still had to stop for aiming and casting.

Figure 25. New UI that included automatic attack. In the right image, the player has a target with the automatic skill.

The functionality of the button whose skill was changed to be automatic was changed from aiming functionality to toggle the aiming indicator on and off by pressing the button. The aiming indicator contained a maximum range indicator and a cone indicator to show the aiming angle. The maximum range indicator showed how far the automatic attack could be launched if the enemy is not inside the cone area. The indicator was enabled by default but it was always temporarily hidden if the player started to aim or cast with other spells. Whether the range indicator was visible or not had no effect on the functionality of the automatic attack, as it was enabled all the time. By this change to the automatic attack button, the UI had to be modified to not confuse players with similar buttons that would work completely differently. The button of the automatic attack was moved from the set of the other skills to the left bottom corner to give it the appearance of a different functionality by separating it from other ability buttons. The new UI is shown in **Figure 25**.

Before adding automatic attack to the gameplay, the actual combat between players took place a large distance. This happened due to players wanting to be safe while casting. If a player would run too close to start casting, the whole enemy team could have a good chance to trap and kill that player. By adding the automatic attacks to the gameplay, players went closer to each other more often.

In the first version, only the other players that the player was targeting were marked with the targeting marker. In the next update, the closest enemy to the player was always marked with a circle under the enemy if the player was not aiming or casting a spell. This provided a fast way to see which of the possible enemies on the screen would be the closest one and, therefore, which enemy would be the target for the spell if the spell were activated. After adding the automatic attack functionality, this had to be changed in a way that, if the player was not aiming or casting a spell, an enemy was marked only if it was

reachable by the automatic attack. This informed the player whether the enemy was close enough to be hit by this attack.

To make targeting enemies and hitting them a little bit easier, a new ‘universal targeting range’ was added to the game and it was the same distance for all heroes. This new targeting range allowed players to aim towards enemies that were further than the skill’s targeting range and closer than the universal targeting range. Even though the ‘universal range’ was not visible for players, this functionality helped players hit easier those enemies which were too far for the skill’s targeting range but who still were running towards the player. Now the player was able to target these kinds of enemies and if the enemies were still running towards the player, they were most likely to hit them. In **Figure 26**, the enemy is marked as the target with the help of ‘universal range’ with a grey target marker. The player was now able to shoot towards the enemy but if the enemy was not moving any closer to the player, the projectile would have not hit its target.

Figure 26. The enemy is marked as the target even though the enemy is at that moment too far to be hit.

By adding universal range, the rules by which the player was marked as a target had to be changed. The priorities for marking players as targets were as follows (smaller number means higher place in priority):

1) Targets found by skills which player activated by itself;
2) Targets found by automatic attack;
3) Targets found by in ‘universal range’ while not aiming;

Targets that were found within universal ranges (in priority list indexes 1 and 3) were marked with more neutral colour to indicate the closest target that was still too far to actually hit it. The addition of universal range brought some balance between manual and

automatic attacks, while also providing better information of whether the target player was almost inside the targeting range or not.

An alternative way of movement was added to the gameplay to give the player ability to move faster for a short time. By double clicking the screen where the movement joystick could have been used (thus the whole screen excluding the actual visible buttons), the player dashed forward in the hero's direction, after which a dashing cooldown was started. This enabled the player to seek enemies or other friendlies faster and also escape from an enemy attack.

A minimap was added to the game as an essential feature. The minimap was placed between the middle and bottom of left side of the screen as seen in **Figure 27**. In portrait mode, the attack direction was designed to be more in a vertical direction than horizontal direction to allow players see better what is happening in the attack direction. To prevent any blocking UI elements in the attack direction, the minimap was located in its current position rather than in the upper left corner. In the minimap, all players were all the time visible. Blue and red triangles were used to show the position of friendlies (and the player itself) and the enemies, respectively. In addition to the minimap, to show the position of the enemies, red triangle indicators were added to the edges of the screen to show to the player in which directions the enemies were located. In the rightmost image of **Figure 27**, the player can see two red triangles in the UI indicating the directions towards enemies.

Figure 27. A minimap was added to the gameplay as an essential feature. The left image shows the position of the minimap in the UI. The middle image shows how the minimap looks during gameplay. Blue and red triangles are used to tell the location of friendlies (or the player itself) and enemies respectively. The right image shows the red triangles on the left side of the screen that indicates the directions towards enemies. The red circles are added to the picture to make the triangles more visible.

To help players understand and learn the game in a better way, the following changes were made: skill button animations and prioritizing skills over movement. When the cooldown of a skill was finished, a flashy particle system was played on top of the button to get the player's attention. If a skill had a potential target in-range, a continuous flashy animation was played around the button. This was made to help the player to understand and know that the skill is ready to be used and a possible target exists. Previously when player was moving, it was not possible to start casting an attack. This would have most likely confused players who would have tried to play the game with two fingers. While moving, they would have probably thought that the skills were not working correctly. This was changed in a way that players could have started aiming and casting skills while moving but it would have also stopped the player. This way players were informed that the two-finger playing was not possible at least to this extent.

4.5 User testing

Several modifications were implemented and tested by the author and/or his team members already in the iteration phase described above. Furthermore, a more thorough test was organized to find answers to perhaps the most important design choices the team identified: namely, is the portrait mode working as intended and whether or not to use skill automation. The user testing session was also held to provide important feedback of how players understood and learned the game functionalities and how to use the user interface.

4.5.1 Setup

During play sessions, two different builds of the game were used: one had automatic skill and the other did not. These builds will be referred to as automatic and non-automatic builds, respectively. **Figure 28** shows a screenshot of these builds. Even though one of the builds had automatic attack and the other did not, the UI looked the same for the both builds. In other words, the build that had no automatic attack had the skill button on the left side of the screen as did the build with automatic attack. The non-automatic button was kept on the left side even though it should have been on the right side just like other manual skills; this was done to keep the UI consistent between the builds, which required that all buttons should be in the same positions in both builds.

The UI in the play sessions was a modified version from the original one that had automatic attack button. This modified UI contained only three skill buttons, thus having one of the buttons removed. The removed button was the top button of the skill buttons on the right side of the screen. The button was removed to make players be less confused by the large number of buttons. In this way, players may learn the other skills better. In all test runs, the test players had always the same hero with the same skills. The button on the left side of the screen (thus the automatic attack button in the automatic build) had a line skill. The bottom and upper skill buttons on the right side had line and high arc skills, respectively.

The gameplay and interviews were recorded to ensure more accurate data collection and to prevent any missing information that could have inadvertently remained unnoticed during manual observations. Ziggi-HD Plus USB Document camera with Presenter software were used to record the play sessions. The testers played the game with OnePlus 3T mobile phone.

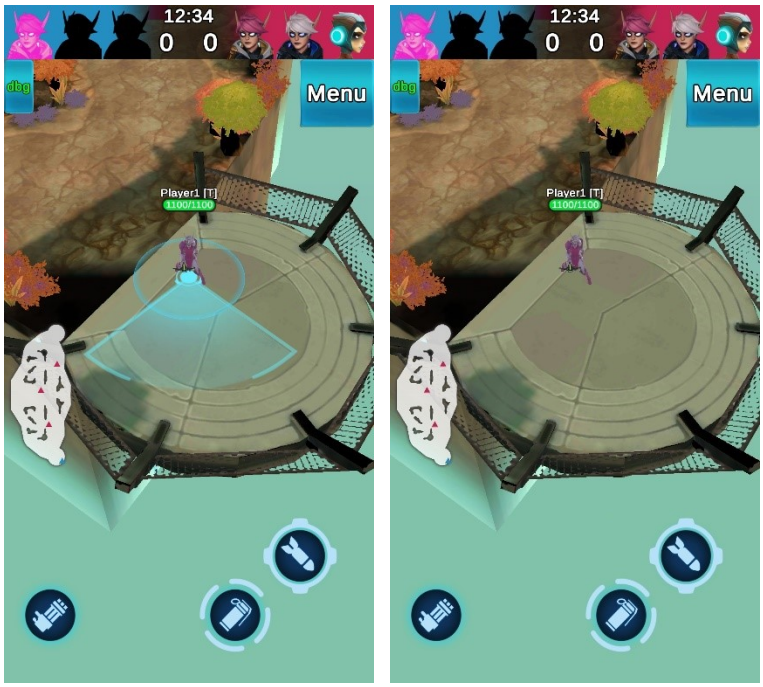

Figure 28. The UI in automatic (left) and non-automatic (right) builds. In the play testing, the time in the middle top started at 3:00.

4.5.2 Play session

The playtesting was started with an introductory speech where the testers were informed about the overall context of the test session. This included a superficial description of the topic, project state, why the tests were held, the permission to record the test session, and finally the structure of the actual test steps. These steps were as follows: gameplay with one of the builds, questions about the gameplay, second gameplay with the other build, the same questions about the gameplay and finally a questionnaire about the gaming habits overall. A session was on average 15 minutes long.

The object of the gameplay was to find and eliminate three enemies in three minutes. The map contained three bot enemies which did not try to shoot the player. The first bot was standing still. The second bot was moving horizontally back and forth. The third bot was running around a small circle. The idea of these kinds of movements was to make it more challenging to eliminate the enemies so that the player would probably act always a little differently and, as a result, possibly use different skills than they were used to.

The questions after each gameplay were the following: 1) What did you think about the controls? 2) What was working? 3) What was hard? 4) What would you change in the game in the first place?

The test session was held with 15 testers who all are worked at Rovio Entertainment Corporation. Every second tester started playing with the non-automatic build and finished with the automatic build, and the other testers did vice versa. The reason for this kind of arrangement was to see if the test results of how players received the automatic vs non-automatic gameplay depended on the order in which players played the builds.

14 of the testers were male and one of them was a female. Ages ranged from 28 to 42 years with an average age of 35 years. All the testers played actively with a mobile phone and 47% with a tablet. Also, 60% of them played games with a PC, 53 % with PlayStation, 20% with Xbox, and 20% with other platforms such as Wii or Switch. The amount of average playtime per week varied greatly among testers. The average playtime among all own platforms per week changed from one hour to 50 hours, while playtimes with a mobile phone/tablet ranged from 0 to 15 hours. Among all testers, the average combined playtime with all platforms was 12 hours and with mobile phone 4 hours per week. Half of the testers played video games 6-7 days per week and the other half 1-5 days per week. The most commonly played games among testers were Angry Birds Friends and Clash Royale, both played by 27% of the testers, followed by Angry Birds Match, Battle Bay and Hearthstone, all which were played by 13% of the testers.

4.5.3 Results

Generally, the different gaming habits were not reflected by considerably different results.

When thinking about mobile games in general, the testers had different opinions between game orientations. The preferred orientations are shown in **Figure 29**. 27% of the testers preferred portrait games. The reasons for this was that in one-handed gameplay, it was easier to hold the device and the player could do other tasks (e.g. pick items and use public transport) more effortlessly while playing on portrait mode than with landscape mode games. 13% of the testers preferred landscape mode games. The reasons for this was the fact that the game view looked larger while played in landscape mode and the player's navigating and action trigger fingers would more likely block the screen less than with portrait games. The remaining 60% of the testers felt that they preferred the orientation that is best for the game genre or the game itself.

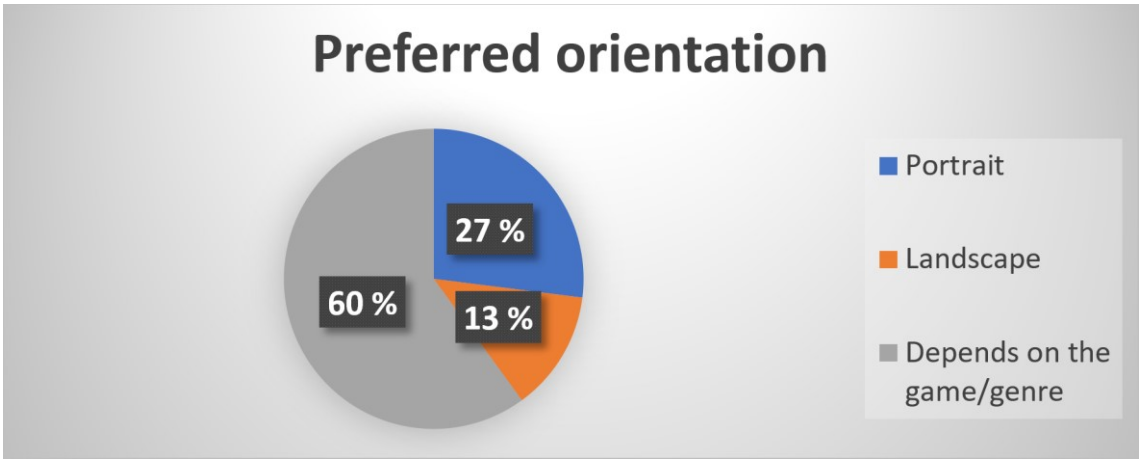

Figure 29. Preferred orientation among the testers.

The testers where asked to give their opinion of the pros and cons in portrait and landscape games. The pros for portrait mode were one-handed gameplay, easier to hold, usually shorter play sessions, and that the gameplay feels more natural and in general because mobile phones are frequently used in portrait mode. The following features were seen as cons: narrow view, the game feels smaller, it is uncomfortable if the game has to played with two fingers and if the game is played with the hand that holds the device, not all corners of the screen could be reached easily. The pros for landscape mode were the wide

view that also made the game screen feel larger, with thumb controls it feels more like a controller, and the landscape mode fits better on tablets than portrait mode. Two-handed gameplay was also seen as a con by forcing the player to use both hands, which prevented one from doing small tasks while playing.

During gameplay, 20% of the testers tried to change the orientation from portrait to landscape mode. The orientation did not change because it was not allowed in the game. Instead of using the joystick for movement, one player tried several times to use tapping to move. While pressing a skill button, two cancel markers were shown to the players. One cancel marker on both sides of the screen. 33% of the testers tried to press these markers with another finger. This did not work because the finger that activated the skill had to be dragged to the cancel marker. This was not clear for the testers. Also, some of them thought that the cancel marker could give some special effects for the ability. 13% of the testers learned immediately how to use the cancel markers when they tried to cancel an ability for the first time.

Data of how the testers played the game is seen in **Figure 30**. 80% of the testers tried to use two different UI features such as movement, ability button, aiming, tapping, or cancel button at the same time. 20% of the testers played the game with one finger. All of these one-handed players played the game with the same finger through the whole gameplay session. However, not one of the testers played the game as it was designed in the first place, which was with the thumb of the hand holding the device.

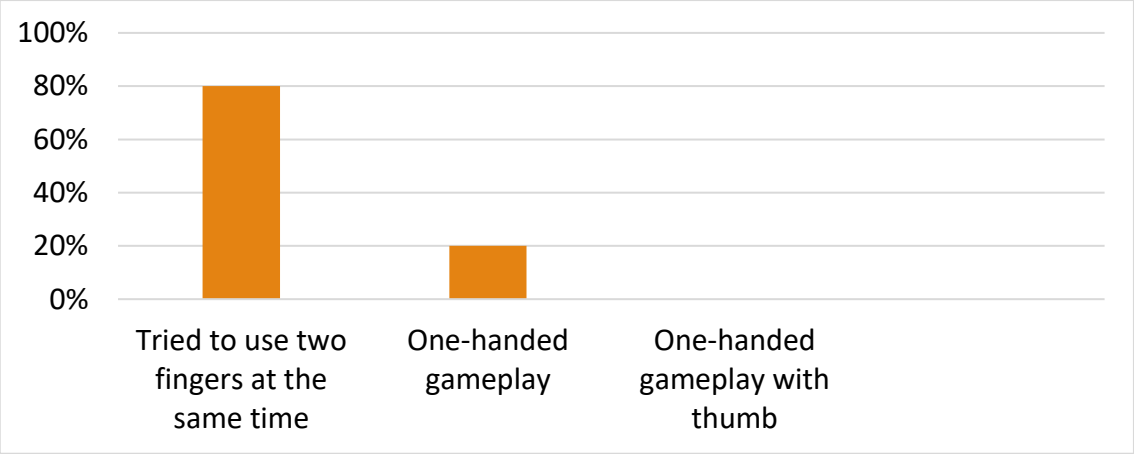

Figure 30. How the testers played the game.

In the automatic build, the auto attack and the UI indicator for showing the range of the attack was clear for all players. Most of them understood its functionality immediately after they found the first enemy and, for some of the players, the auto attack was not clear at first but they learned how to use it before killing the first enemy. In the automatic build, the left ability button was only used for enabling and disabling the UI indicator of the attack area of the auto attack. This was fully clear for only 7% of the players. This button created confusion in its use. Most of the players thought it would activate and deactivate the usage of the automatic weapon, while some of the players never even tried it.

Differences in learning manual aiming divided the players into four groups depending how long it took to learn it. The groups are illustrated in **Figure 31**. 7% of the players learned manual aiming before they caused any damage to enemies. 26% of the players

learned to use the manual aiming before killing any enemies. 20% of the players learned the aiming after killing the first enemy and other 20% learned aiming after killing the second or third enemy. The remaining 27% of the players never learned how to aim manually.

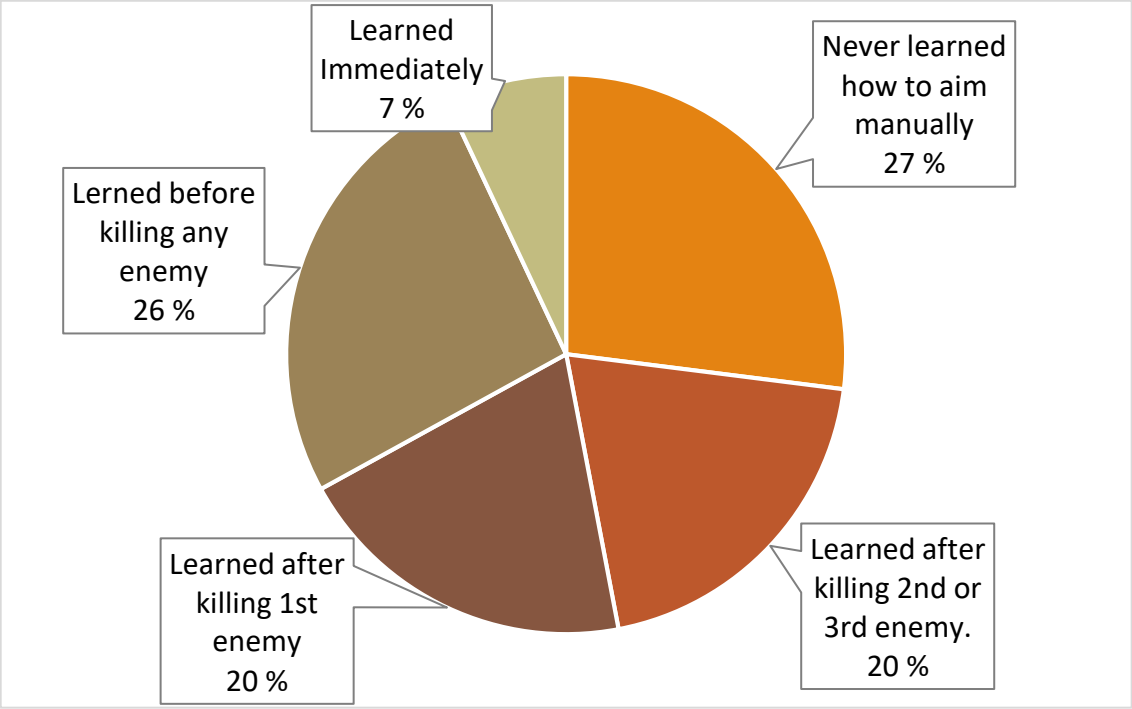

Figure 31. How the players learned to use manual aiming.

During gameplay, 60% of the players tried to shoot enemies by tapping on them before or after starting to cast a spell. While playing the non-automatic build after the automatic build, one of the players tried to keep pressing the left ability button to shoot automatically towards the enemy. This was most likely affected by the learned way of playing from the automatic build and the player tried to get the automatic shooting to work by holding the ability button.

The goal of the game was to find and kill three enemies in three minutes. If the game crashed or had a game-breaking glitch, it was restarted. If the game crashed or had a game-breaking glitch for a second time in the same build, and if the player had killed at least three enemies overall during the last two gameplays, the game was not restarted again. In these cases, the result of the player winning or losing was ignored. Only one gameplay winning/losing result was ignored. From a total of 29 valid gameplays that ended by the killing of all three enemies or having the time running out, 28 games were won and only one game was lost.

After each session of gameplay with the automatic and non-automatic builds, the testers gave the team their perception of how they felt the controls of the game. Overall, the testers found moving to be working well and following the joystick was found to be useful. However, the game UI did not give the players any hint of how to start moving. Some players were not sure if they were allowed to move or where the movement happens. The indicators were visible only during the actual movement while the player dragged a finger along the screen of the device. Also, the moving speed was seen to be

too slow, at least for an offline fight against bots. Some players would have preferred a tap-to-move mechanism over the virtual joystick.

The overall UI placements were not received well. One reason for this is the blocking effect of the moving finger. The finger often blocked either the minimap and the left skill button or the skill buttons on the right side. The minimap was also seen to be too low. Furthermore, raising the minimap higher would have prevented the minimap from being blocked by a finger. The button placement and the buttons themselves were seen as confusing in both builds. In the non-automatic build, having one of the ability buttons on the left side let it exposed to blocking by the finger. In the automatic build, the left ability button, which was now only for showing and hiding the auto-attack target range, was not clear. In both builds, the button differences and how the buttons worked were not clear to everyone. Freedom of aiming and aim assist, which locked a target automatically or after manually aiming if an enemy was close enough, was a popular feature. Nevertheless, for some players the manual aiming was hard to learn. Also, an aiming type image in the ability button was requested.

The functionality of only moving or shooting while standing received criticism. Most players did not see this feature interesting but boring instead. The stopping while aiming and shooting made the game feel disrupted and it made the players stop next to the enemies and spamming the ability button immediately when the ability cooldown was over. By enabling auto attack, players were moving more while shooting enemies, although the majority of players still just stopped next to an enemy to kill it faster.

When comparing the user experience between automatic and non-automatic builds, few key points were observed. In manual gameplay, the players did not like the left ability button that used to be the auto attack in the automatic build. Now the players had to keep pressing the button continuously for two reasons: firstly, the other abilities had much longer cooldown times and secondly, the left skill button had fast cooldown but low damage. This button spamming was seen as one of the worst features in the game. Although the players learned the abilities better in manual gameplay and they felt that they had stronger control in the fights and more to do. In the automatic build, the auto attack was seen at the same time as a really good feature but it also made the game really boring against the bots by taking the control of the fight away from the player. Nevertheless, the auto attack disabled the need for repeatedly pressing the ability button but it also allowed easier gameplay against bots by allowing the auto attack to kill the enemies by itself eventually. The direction of the shooting area of the auto attack was seen as problematic. First, the area was quite small and the player needed to go really close to the enemy, while even small turns easily moved the shooting area away from the enemies. Also, shooting automatically only towards the running direction was seen as a possibly bad idea for real player-versus-player gameplay, since to use the auto attack the player would had to do a ‘suicide’ attack. However, this last comment was only a hypothetical vision of the gameplay and how the auto attack might work there.

Overall, it was also mentioned that showing enemies all time in the minimap was not natural. The enemies should have been visible only when you see them or get closer. The game was in the end hard to learn and making a portrait mode was seen as risky, since MOBA games has been mostly made for landscape mode. Therefore, making a portrait

game, especially in a genre that only has landscape games almost exclusively, would be making a game against standards.

After asking the testers what was working in the game, only a few common features popped up. The most commonly mentioned features were movement (60%) and the freedom of the movement joystick (20%), auto attack (27%), minimap (13%), the controls in overall (13%) and how the game looked (13%). Other shooting features that were shown as working for some testers were the icons for the weapons, manual aiming, aiming assist, aiming indicators, multiple skills to use, and homing missiles. Also, portrait mode and the learning curve in the game was mentioned to be working. However, 20% of the testers did not see anything good in the non-automatic gameplay.

Even though overall the controls and aiming were seen as working features, those were also seen as difficult to learn. The testers found that switching between movement and shooting was hard and not a working functionality. Also, the overall slow tempo in the game was not seen as a positive element. Movement were slow, manual aiming took time, and finally the player had to wait before the spell was cast. Therefore, movement, aiming, and shooting were not seen as intuitive by some players. The players also had difficulties in learning and understanding the attacks and how to use the skills correctly and in the best way. Aiming was said to felt clunky and most players had difficulties learning to use aiming with the high arc skill, which was a grenade in the gameplay. During aiming, the rules of locking targets or when shots were homing towards the targets, were not always clear. Also, killing enemies with manual attacks was thought to be hard. 20% of the players mentioned that cancelling aiming was not something they learned and it felt hard or not obvious. For one player, the functionality of how auto attack worked was not optimal for MOBA games. Some players mentioned that while playing the game, the active finger blocked too much of the screen. One player learned to activate the dash but its direction was not understood and it was therefore seen as weird.

The last question after each gameplay was what the tester would change in the game in the first place. The most wanted change by 33% of the testers was to reorganize skill buttons better on the UI to get them on the same side of the screen to allow for easier movement on the left side of the screen and to prevent any buttons from being blocked by the finger while moving. For the same reason, 27% of the testers wanted to move the minimap higher on the screen up to the top. Without any further ideas or examples, simpler skills were wanted, how aiming works modified, changing movement, and making shooting more intuitive. To get a real taste of challenge and battle, enemy bots should shoot back at the player. Movement speed was wanted to be faster and also making the player move faster when dragging the finger further. To ensure better flow for the gameplay, two-finger gameplay would have been brought to the game to allow moving and shooting at the same time. The majority of the players preferred auto attack in the gameplay and they would have kept it, but maybe also made it shoot with a longer cooldown to make it less powerful. 13% of the players would have preferred a tap-to-shoot mechanism in the game. To make the ability button functionalities clearer, some kind of popup or visualization of the ability was desired by 20% of the testers. Also, an idea of a single shooting button was suggested. In that concept, the player could change the active skill by aiming further, dragging the finger further while aiming, or by aiming a longer time. Changing the game to landscape mode and allowing two-finger gameplay were also recommended by 13% of the testers.

4.5.4 Analysis of the results

The main purpose of the user testing was to learn how players learn the gameplay mechanics. The main questions were: “is the movement clear?”, “do testers play the game with the thumb of the hand holding the device, or do they play it with another finger, or do they play it using multiple fingers?”, “do players learn how to use skills automatically or manually?” and “is the current portrait gameplay success, failure or something between them?”.

The first observation from the play sessions related to the way players played the game. None of the players played the game as it was originally intended. The game was supposed to be played with the thumb of the hand holding the device but the players did not do so. However, some of the players still played the game with one finger. The reason for the two-finger gameplay habits was most likely affected by the game itself and the UI. The game is a 3D MOBA game where players shoot and kill the opposing team members in real-time. These kinds of games are usually played with two fingers to provide faster and more accurate gameplay without any breaks. Players also were not used to stopping while performing actions, and this way the developer team tried to break the standard of MOBA gameplay. To make the gameplay more natural for one-finger gameplay and make the use of other fingers meaningless, some large changes to the game functionalities and UI would need to be made. To fully disable any advantage of using two fingers while playing, UI buttons should maybe be removed. This way players would only interact with the whole screen. Different features such as movement, switching abilities, aiming, and shooting would happen by an alternative method, such as by dragging, tapping, or through gestures.

The movement joystick and its freedom to be used anywhere on the screen area was praised. However, it was observed in the play session that the follower joystick was too sensitive. If the player moved the finger outside of the joystick area, the joystick started to follow the finger by trying to keep the finger all the time inside the movement joystick. Now the player could not drag the finger back where it was without moving back with the hero character or releasing the finger and creating a new movement joystick by touching the screen again. This follower joystick which was too sensitive affected the movement finger to move in a wider area inside the screen area and to block more buttons and the environment. The testers felt that the finger blocked buttons only because of the button positions, but the movement joystick actually had a negative effect by moving the joystick towards the finger even if the player moved only a little bit outside of the joystick. This accidentally forced the players to use the movement joystick in a non-optimal way. A more optimal solution for the movement joystick would still allow it to move towards the finger but in a stricter way. The joystick would follow the finger when the finger has reached a distance that is further than the allowed maximum distance ‘d_max’. In the play session, this ‘d_max’ had the same length as the radius of the movement joystick. By increasing the value of ‘d_max’—for example, by doubling its value—it would most likely make the movement joystick to move less during the gameplay. Alternatively, the joystick could be made larger with the old rules of following the finger. With a larger movement joystick, players would not gross the outline of the joystick as easily. In the other hand, the larger movement joystick could easier block more of the visible play area.

To prevent the uncertainty of whether a player is allowed to move or not, and how a player could move in the game, the movement joystick should be visible at all the times. Whenever the movement joystick would not be used, it would jump back to its default position to clear the game area (to prevent cases where the movement joystick would otherwise block some parts of the screen) and to indicate to the player that the movement joystick is allowed to be used again.

In both builds, the placements of the UI elements were not fully liked. Keeping the UI look the same in the both builds kept the UI unchanged, but it also had negative consequences. The left ability button looked the same in the both builds but it had different meaning. Also, in the non-automatic build, this situated one of the ability buttons on the opposite side as the other buttons. This caused some confusion in the players and it was predictable. Keeping the similar working buttons on the same side would make the UI more consistent and, in the case of the non-automatic build, this would have decreased the amount of the buttons being blocked by a moving finger. In gameplay, often one of the sides was blocked by the moving finger.

Some players tried to tap to move or tap on an enemy to shoot them. The reasons for trying to tap to move may be curious to try whether it would also work, or the missing movement joystick while not moving might have given an assume that there is also alternative way of movement. The behavior to tap enemies to shoot towards them can be attributed to a number of causes. One cause can be due to a misunderstanding of the aiming and target-selection rules. In gameplay, there was not a situation where two enemies could be in range of the player at the same time. Therefore, the players could not learn how it works with multiple enemies. However, this should not have an effect on how players learn to aim and shoot towards a single enemy. Some players learned accidentally incorrectly to tap on enemies to shoot them. After the player had pressed and released a skill button without dragging it, the skill chose the target automatically and started the casting time. During the casting time, these players tapped on the enemy, after which the casting time ended in a fraction of second and shot the enemy. Instead of learning that the casting time is intended to be a delay for an attack, these players thought it was time to start tapping on enemies.

One reason for why the cancel buttons was confusing was its grey color and far distance from the skill buttons. To be able to cancel an ability, the finger that started the skill had to be dragged far away, thus potentially making it feel incorrect. By rearranging the skill buttons on the same side, the cancel button on the opposite side could be removed as unnecessary. To make the functionality of the cancel button clearer and improve the usability, the button should be brought nearer to the skill buttons and the image of the button improved to look more like a cancelling action.

Although the minimap was placed under the middle point of the left side of the screen to prevent blocking any UI elements in the attack direction, the players would have preferred it to be placed on the top left corner of the screen. The minimap was seen to be too hidden due to being too low and it was blocked multiple times during gameplay. As the game is in its current state, the minimap should be placed higher. However, by changing the gameplay mechanism to work better with one-handed playing, the minimap location could be more or less workable.

In the non-automatic build, the left skill button was seen as one of the worst features in the game because it had a very fast cooldown period but too low damage output. The result of combining these two last mentioned features is unpleasant in terms of user experience where players have to continuously press and release the skill button. The reason for this weak but fast skill was to bring a default skill that players could use as a default while other powerful skills were in cooldown. But due to the game mechanism, which prevents movement and shooting at the same time, this kind of skill that needs to be used multiple times to have some effect does not bring anything worthy to the game.

Auto attack was seen as a good feature by most of the players but some of them found it at the same time to be bad. The mission in the gameplay during the play session was easy. The players had to find and kill three dummy bots that did not shoot the players or try to escape. One of the bots was not moving and the two others moved in a small pattern: back and forth or in a circle. In this gameplay, it was not necessary to find good strategies to kill the enemies but just keep shooting them as fast as possible. Shooting a target that is not doing anything interesting or fighting against the player is not very fun. The reason for the popularity of the auto attack may have been these aforementioned aspects. With auto attack, the player could have finished the game faster without repeating the same attack over and over again but could still try the other skills to explore their uses or simply to speed up the game progress. However, in the online player-versus-player game mode, the situation would be the opposite. The enemies would run towards the player or try to escape and shoot back at the player. These circumstances would heavily affect the gameplay feel. However, the test session was held with dummy bots to keep the gameplay consistent for all players and to analyze how they learn the basic game mechanisms.

27% of the players did not learn to use manual aiming. These players never tried to use them even accidentally. Although these players tried to attack the enemies beyond the auto-attack range and the maximum targeting range of the current skill, they never tried to drag the finger after pressing a skill button. Therefore, this feature should be explained to players more clearly with indicators in the skill button or by creating a new joystick over the skill button after it has been pressed to indicate that the players could somehow use the new joystick with the skill or by just adding tutorials to the starting of the game to teach the players to use different game mechanisms such as manual aiming. Furthermore, the rules for locking the enemy as a target and where the spell would actually fly was not clear to everyone. The enemy was locked and targeted automatically if the enemy was inside the maximum shooting range circle UI indicator and if the player did not start aiming manually. In these cases, the player was turning and aiming towards the enemy automatically when the enemy was moving. The player still kept the enemy as locked even if it walked outside of the targeting range while the player was targeting it. When the player started to aim manually with the line skill, the closest enemy to the targeting line was selected as a target when the player released the finger. If no valid target was found, the spell was shot towards the aiming direction—otherwise the player acted like it would have automatically chosen the enemy as a target and hence the player kept turning and aiming toward the enemy during the casting time as well. In addition to this, the spells on auto attack and line skills were homing toward the target enemy. These functionalities would help to hit other players in the online game mode while they would be moving when the player is aiming, casting, and finally shooting. Nonetheless, these rules were quite complicated for new players and for this reason the players did not fully understand in which cases the enemy was locked as a target and where the spells would

have actually flown. These rules are hard to teach by UI and a tutorial for this would be needed. Alternatively, these rules could be simplified in a way that no tutorial would be needed, but it could again have a large effect on gameplay.

4.6 Alternative ideas

During the project and while iterating the UI, some other ideas was tried or analyzed, and these were tap-to-move, automatic movement, a movement helper, and stamina for movement.

4.6.1 Tap-to-move

Tap-to-move was considered for movement in place of joystick. However, this had two features that the team did not want in the gameplay. The first was continuous tapping on the screen. The view of the game area is quite small and a player would not be able to tap far away; this meant instead that players should be tapping every second where they would like to be moved while blocking some parts of the screen. Also, during encounters with other players, the players would probably change the moving direction quite often, which would lead to fast multiple tapping on the screen. The second reason for not wanting to use the tap-to-move function was the overall feeling of it and how it would affect players' experiences during gameplay. While using the movement joystick, players can immerse themselves and think that they are the hero in the game, while in the tap-to-move gameplay players could easily have the feeling that they are just giving orders to the hero. The difference in immersion during gameplay between these two alternatives can be small, but it can also be one of the reasons to keep playing the game or not.

4.6.2 Automatic movement

With automatic movement, the idea was to enable the players to start moving automatically. This could have been started by making a fast swipe and then the player would keep walking in a given direction until the player started manual movement. A player could also tap on the desired position in the game area and player would walk there automatically by using the shortest path. This would allow the player to move and shoot at the same time. However, this would greatly advantage automatic movement over manual movement because only automatic movement would allow shooting while walking. In this kind of gameplay, players would mostly use automatic movement and in some special cases manual movement. If the automatic movement could be used easily with tap-to-move, no manual movement would be needed. At this point, the movement joystick would be thrown away and only tap-to-move would be used in the game. Alternatively, if the shooting would be enabled while walking manually, the game would be played with two fingers and therefore it would be better to be made in landscape mode to ensure more space for the fingers.

4.6.3 Movement helper

Before adding the casting time to gameplay, a movement helper functionality was tried. Let us mark 'd_max' as the maximum shooting range. If the enemy was further than 'd_max' but less or equal to 1.2 * 'd_max' from the player, after the player has released a skill button to start a spell, the player started to move towards the closest enemy in a given direction for one second or until the enemy was in range of 'd_max'. If the enemy was finally within range, the spell was then started. Otherwise the player would have just

stopped after walking. This feature was working well in the gameplay tests among the developer team, but it could not have been used well after the casting time functionality was added to the gameplay.

4.6.4 Movement stamina

Stamina was a fast experiment in the early stages of the project. It was meant to restrict the movement of the player. While players could only move or aim and shoot, the shooting had to either have some helpers or the movement had to be made stricter. With movement stamina, players could not move forever because the stamina was used while walking. The stamina could be gained back by standing still and optionally aiming and shooting. This feature made switching between movement and shooting more interesting and more tactical. When the stamina was running out, players had to start to prepare for that by getting cover and starting to aim towards possible threats to protect themselves until there was enough stamina again. However, adding a movement stamina and making movement even more stricter was not seen as a great or interesting feature for a MOBA game.

4.7 Analysis of UI and UX in portrait mode

The working one-finger gameplay was the primary goal for the project. However, after multiple tweaks and additional new features that could have made the gameplay feel better, the game was too slow for the team's requirements and implementation. The first step that got the team to go in the wrong direction was probably the requirements for the game. Making a one-finger portrait game from a two-finger portrait mode while keeping all the main aspects, mechanisms, and gameplay flow would not happen without sacrifices. Making the game work by tap-to-move and tap-to-attack in the same way as in Vainglory (in chapter 3.5.3) would have most likely made a huge improvement for the gameplay flow, but this feature was not wanted for the team's game as discussed in the chapter 4.6.1. Due to the results of user testing and the discontinuity of gameplay flow, there is no easy solution to transfer the game to work well in a one-finger portrait game. Making the game in portrait mode and allowing two-finger gameplay like in the portrait mode version of Brawl Stars (described in chapter 3.5.8), the problem of the gameplay flow by switching between movement and shooting would have been fixed. However, playing a portrait game with two fingers can lead to a bad gaming experience when the screen of the mobile device is too small. The fingers would then easily block too much of the screen. Also, if the game could be played with two fingers, then the orientation should be changed to landscape. Overall, at this stage, the team declared the portrait one-finger gameplay to be a failure. The project was continued in landscape mode.

4.8 Changing game orientation to landscape

At the start of designing a gameplay UI for landscape mode, many features were designed again. One of the biggest initial changes was the ability of being able to move and shoot at the same time. Also, after this change the casting time was not required and it was removed from the game to prevent unwanted breaks in the gameplay.

Another initial change was modifications to the movement joystick. As the user-testing results brought up, the missing movement joystick created some misunderstandings about the movement. To prevent any confusion about how movement works, the movement

joystick was shown to the player in the bottom left corner in the screen. The movement joystick was still able to be used anywhere within the screen area. Another point that was noticed during user testing was the too sensitive follower functionality in the movement joystick. In the new version, the movement joystick started to follow the finger when the finger's distance from the movement joystick's center was the length of the diameter of the joystick.

For trying to make the game easier to understand and to get a distinctive feature from other MOBA games such as those analyzed in chapter 3.5, new rules of how skills were used were added. Every hero had two guns, one ability, and one ultimate skill. The player always had one of the guns or ultimate skill as the active weapon. The activation and aiming of both guns and the ultimate skill was done from the same shooting button and it was always applied to the active weapon.

Figure 32. The UI layout of the landscape mode.

When changing the game orientation from portrait to landscape and to fit the old and new buttons, the UI layout had to be organized again to make the visualization clearer. In **Figure 32**, the new layout of the game is shown. The shoot button that handles the activation and aiming of the guns and the ultimate skill was located at the right bottom of the screen. The primary weapon, ultimate skill, and ability button were in this order around the shoot button. The secondary weapon was located on the left side of the primary weapon. The active weapon was marked with a white circle around the active weapon button and with a white line that connected the active weapon button with the shoot button. In addition, different aiming types, maximum shooting ranges, and ammo counts that differed on different weapons were added. The active weapon also affected the game view. The game camera zoomed in and out depending of the maximum targeting range of the active weapon. Therefore, with some weapons the player was able to see further than with some other guns. This feature is illustrated in **Figure 33**. Also, the minimap was moved to be in the top left corner and team information and score on the top right corner of the screen. Furthermore, the leftmost cancel skill button was removed from the

game because the skills were located on the right side and dragging a finger through the whole screen would not have made sense ergonomically.

Figure 33. The camera zooms in and out depending of the maximum shooting range of the active weapon. The game view is larger while aiming with a missile (below) than shotgun (above). The missile shows the width of the projectile by the width of the aiming line. The missile is at the moment reloading. Even though the reloading is active, the player can shoot with the active weapon because it still has two shots available.

To encourage players to switch weapons more actively, the weapons were given ammo. After every shot, there was a cooldown in the same way as previously with the skills. After the clip of the weapon was empty, the weapon could not be used before it was reloaded. The reload worked the same was as cooldown but it was a longer period and

the reload progress was shown above the player. The reload icon can be seen in the below image of **Figure 33**. Reloading started when player was long enough without shooting with the active weapon or when the clip got empty. If the weapon had ammo, the reload could be cancelled by shooting with the gun. The ammo is shown in three different areas at the same time. First, the ammo indicator location was around the weapon button as yellow circle and divided into as many parts as there was rounds of ammunition. Similarly, the shoot button used this same method for showing ammo for the active weapon. The ammo that was shown in the shoot button belonged always to the active weapon. The last ammo indicator was below the health bar. A yellow line was divided into as many parts as there was rounds of ammunition. When player used ammo, the ammo indicators in the buttons turned grey, and in the ammo line, they turned black.

Even though the ultimate skill was seen as a weapon, its cooldown system differed from the guns. To be able to use the ultimate weapon, its energy had to be fully charged. At the start of the game the energy was empty. Every second of gameplay charged the ultimate skill little by little. Also, killing enemy players gave large bonus energy to the ultimate skill. After the energy was fully charged, the ultimate skill could have been used. After the usage of the ultimate skill, its energy was again empty. The charge level of this skill was shown with percentages in the skill button. When the skill was ready to be used, it was highlighted.

The weapons brought some clarity while being play tested by the author and the team. Previously in the automatic build of the portrait mode, the players were using automatic attack and three different skills. Now the players were mostly using only two different weapons and sometimes the ultimate skill and an ability. Also, the guns were balanced in a way that the player could play well by using only one of the weapons. This could not be the case in the portrait mode where skills differed hugely by the amount of damage and long cooldowns.

Unlike in the portrait mode, aiming was allowed during cooldown or when reloading was active. In these cases, the aiming indicator was colored red to indicate that shooting is not allowed yet. Also, targeting rules were changed from the portrait mode. The universal targeting range was removed from the game and enemies that were under automatic targeting—or while not aiming, the enemy with least HP within automatic targeting range for the active weapon—were marked as targets. To make the target more visible, a big red triangle was added above the target enemy's health bar and it can be seen in the below image of **Figure 33**. Automatic targeting was enabled with all guns and it was used while manual aiming was not started as in the portrait mode. When in automatic targeting mode, only the maximum range circle indicator was shown to the player.

By allowing players to aim during cooldown or when reloading was in progress, players could start aiming towards the target directions and prepare to shoot. This feature gave players more time to aim without losing any valuable moments, which could have happened if the players would have been allowed to aim after the cooldown was over.

After allowing shooting and moving at the same time, the aiming helper functionality in the line skill prevented the player from predicting the target enemy's movement and from shooting slightly in front of the enemy. For this reasoning, the aiming assist while shooting manually was removed. To illustrate better how wide the bullets will be effective

with the line skill, the line was widened to match the width of the projectiles. This can be seen in the below image of **Figure 33**. These changes helped players to more easily understand how wide the effect of the projectile would be, and the fully free manual aiming with the line skill gave players more freedom in shooting and it was seen as a positive feature.

Projectiles that damaged enemies with a wider area than the width of the projectile had a circle indicator attached to the projectile to better inform the player about the affected area. While aiming with this type of guns or ultimate skills, the area-of-effect circle was shown at the end of the line indicator. With other types of aiming it was not shown. After shooting these kinds of projectiles the area-of-effect circle moved along the projectile. In the bottom image of **Figure 33**, the area-of-effect circle can be seen at the end of the line indicator.

Some weapons, skills, and abilities were able to drop items to the ground. One of these items was a banner that gave health to one's team members. The affected area was shown to the player by a circle indicator that was only visible to one's own team members. When the hero received health points, a flashy green particle system was played around the hero to make the positive effect more obvious to the player. The other droppable item was a turret that shot automatically at players on the opposite team when the enemies were inside a predefined targeting range. The targeting area was shown as a red circle to the enemies.

At this point, the players had only a few reasons for aiming manually with weapons that allowed automatic targeting. The manual aiming was good only for choosing your target by yourself. Nevertheless, this made the shooting slightly slower than automatic shooting and in regular cases players did not care which enemies they hit, especially when the automatic targeting selected targets with the lowest amount of HP. This was changed in some weapons by allowing the projectile to fly further than the player could automatically target. The automatic targeting range was shown to the player by the maximum range circle indicator. The length of the line or cone indicator in weapon or ultimate skills

Figure 34. The length of the line tells that the projectile will fly further than the player could automatically target enemies.

indicated how far the projectile would actually fly when its shot either automatically or manually. In **Figure 34**, the player is able shoot further than the player could automatically target an enemy. This feature made the player shoot manually with the guns and ultimate skills while not being in close combat against enemies.

While in combat with enemy players, it was still hard to try to escape combat and survive. To make escape more possible in some cases, some bigger weapons made the player move slower while aiming or shortly after firing. This functionality gave the player two choices: fight against the enemy until one dies or try to escape without shooting and maybe survive the combat situation alive.

4.9 User testing for landscape mode

Several modifications were implemented and tested by the author and/or team members while changing the orientation from portrait to landscape mode as described in the previous chapter. Furthermore, a more thorough test was organized to answer the question whether the game had gone in a better direction. A user-testing session was also held to provide important feedback about how players understand and learn the new game functionalities and how to use the new user interface.

4.9.1 Setup

To get more reliable data on whether the project had gone in a better direction by changing the game orientation, the same group of testers that were testing the portrait mode were invited to this new test session. Even though the testers had more or less learned the game mechanism in portrait mode, it was not supposed to give them too much information for the landscape mode as the game rules were significantly changed.

The gameplay and the interviews were recorded to ensure more accurate data collection and to prevent any missing information that could have inadvertently remained unnoticed during manual observations. A Ziggi-HD Plus USB Document camera with Presenter software was used to record the play sessions. The testers played the game with OnePlus 3T mobile phone.

4.9.2 Play session

The playtesting was started with an introductory speech where the testers were informed of overall context of the test session. This included a superficial description of the topic, project state, why the tests were held, permission to record the test session, and finally the structure of the actual test steps. These steps were the following: one gameplay session after which questions about the gameplay were to be asked. During the question phase, the testers were allowed to start new matches.

The questions asked after gameplay were the following: 1) What did you think about the controls? 2) What was working? 3) What was hard? 4) What would you change in the game in the first place? After this, the testers were given the automatic build of the old portrait mode game to be played while answering the final question, which was: Which game, the automatic build on portrait mode or the new version on landscape, felt better and why? The overall play session was on average 5 to 10 minutes long.

The objective of the gameplay was the same as previously: to find and eliminate three enemies in three minutes. The map contained three bot enemies that did not try shoot the player. The first bot was standing still. The second bot was moving horizontally back and forth. The third bot was running around in a small circle. In other words, the bots were acting the same way as in the portrait-mode testing. Also, the map was the same, but the camera was rotated 90° to ensure a better gameplay angle for landscape game.

4.9.3 Results

The following percentages in this chapter indicates how many of the players gave the corresponding opinion during the play session. For example, if 20% of the players said that they liked feature X, it does not mean that the other 80 % of the players did not like it. In this case, only the 20% players gave their opinion.

During the gameplay session, it became clear that without any kind of tutorial the UI is not immediately and correctly learned. Nevertheless, the UI was learned before the end of the game. All the players won the game by finding and killing the three enemies within three minutes.

During gameplay some interesting behaviors were noticed. These observations are illustrated in **Figure 35**. During gameplay, 33% of the players tried to tap on the enemy just like in the portrait mode. Unlike in the portrait game where the skill buttons were used to aim, in the landscape mode, many of the buttons were meant only to change a weapon and the shoot button was used for aiming. Despite this, 33% players tried to aim by trying to drag the weapon buttons. In the majority of the games, the players learned only one way of aiming with the guns, either the manual or automatic method. 40% of the players used only manual aiming and 33% of the players used only automatic aiming. Therefore, only 27% of the players used both manual and automatic aiming. Also, the majority of the players had problems with understanding weapon-switching at the beginning of the game.

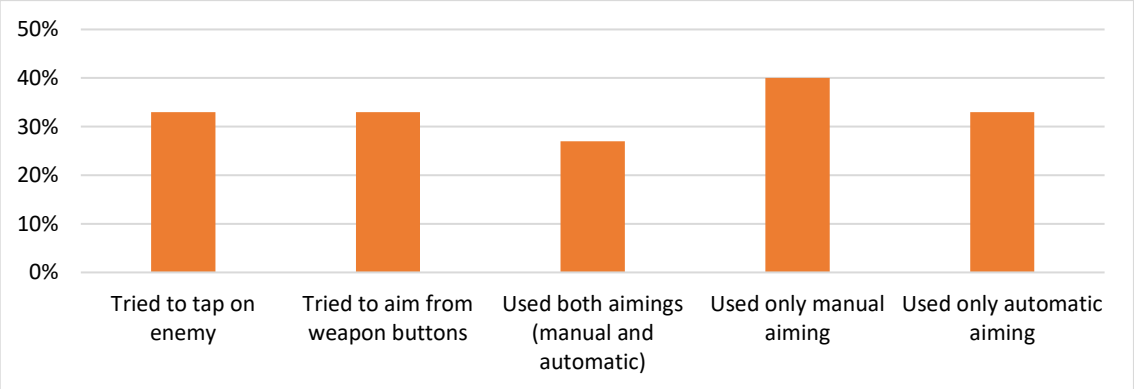

Figure 35. Observations during gameplay.

After gameplay, the players were asked their opinion about the controls in the game. The first thing 33% of the testers said immediately was the fact that the controls were much better than in the portrait mode. Other common features were the overall better feeling of the controls and the movement.

When asking what was working in the game, some common key features were found. First, 33% of the players mentioned that they liked the controls and the fact that the movement joystick was not locked into a certain position, and rather, was centered into the point where the player started to drag finger for movement. 20% of players mentioned the moving and shooting at the same time as a really good feature and that it was a needed feature in the game. 33% of the players found the camera zooming according to the current active weapon as a working feature. Other working features were overall camera distance and angle, the minimap's new position on the screen, and the game graphics.

The hardest feature in the game was switching weapons. 67% of the players had more or less problems with understanding how the switching mechanism worked. Some of the players tried to drag the weapon buttons and some of them kept pressing the same weapon button over and over again and were left confused because nothing happened in the game after pressing the weapon button. However, all of the players learned how the switching worked before the game ended. Another hard feature in the game was aiming, which was mentioned by 13% of the players. The difficulty of aiming came with the confusion of which button was used for aiming and from the high sensitivity of the aiming joystick. These players tried to first tap on enemies to attack them and tried to aim with the weapon buttons. 13% of the players mentioned that the UI buttons were too similar to each other, which made understanding the UI much harder.

The players were mostly happy with the game at the end of the play session and they had only a few ideas on how they would change the game. 20% of the players found the layout of the buttons sub-optimal because the buttons were a little bit too spread out from the shooting button. 13% of the players also mentioned that the color of the minimap was bad because it did not stand out much from the game environment and they would change it to be more visible throughout the whole game. The players who did not like the similarity of the buttons would change them to look the same if and only if their functionalities are the same, thus separating differently working buttons with distinct looks.

All of the players found the landscape game to be better than the automatic build of the portrait mode. There were numerous reasons for this. The most common positive aspects of the landscape mode compared to the portrait mode were that, in the landscape mode, there was more space for fingers (47%), the game view felt much larger (40%), and the more natural controls (20%). Some of the players saw some potential in the portrait mode if the new game mechanisms would be brought into it, or if the game worked as a turn-based game. However, the portrait mode was seen as more a limited orientation and the landscape mode was more convenient for two-handed gameplay.

13% of the players mentioned that the new mechanics that allowed moving and shooting were much better than in the portrait-mode build where the player had to stop while shooting. Also, the too sensitive movement joystick in the portrait mode was seen as bad.

The overall feeling of the new controls in the game was positive among the players.

4.9.4 Analysis of the results

The main purpose of the new user testing was to observe how players would learn the new gameplay mechanic and conclude whether the game had gone in a better direction by changing the game orientation and mechanics. The main questions were "do players

learn how to switch weapons and use them automatically and manually", "is the current landscape gameplay better than the portrait mode", and most importantly, "is the landscape gameplay success or failure".

As the new UI was made to look more traditional compared to other MOBA games in chapter 3.5, the results were expected to be better than in the portrait mode testing. The movement joystick was now visible all the time to players and its functionality was clear to all players. Also, the new rules for the following feature of the movement joystick was seen as better.

Switching weapons was the most misunderstood feature in the game. The players pressed the same weapon button multiple times or tried to drag the weapon button without any change. The causes for this behavior can have multiple reasons. The players may have tried to use the weapons by dragging them just like in the portrait mode. Also, other MOBA games such as the games analyzed in chapter 3.5 utilized skills by dragging the specific button. Another reason can be the UI itself. The shoot button, weapon buttons, ultimate skill button, and the ability button looked the same with small modifications, such as sizes, different icons, and the ability button did not have any ammo. Having the buttons look similar but work very differently may have caused some misunderstandings at the beginning of the game.

In future updates, the grouping of the buttons has to be made visually clearer for players. The differences between the shoot button, the weapon switch buttons, including the ultimate skill and the ability button, should be made obvious.

Many of the players liked the new controls more than the controls in the portrait game. The reason for this was mostly the two-handed controls, which allowed shooting and moving at the same time. Furthermore, the fingers blocked less space on the screen in landscape mode than in portrait mode. On the portrait mode, the fingers easily blocked a lot of the half bottom part of the screen, whereas in the landscape mode the fingers were located at the bottom edges of the screen.

Some players saw potential in the portrait mode by applying the same control mechanics from the landscape mode. On the other hand, this would make the portrait game be played with two hands and it would lead to more screen blocking by the fingers. Also, with smaller screen sizes, the two fingers would easily collide with each other.

The overall result of the testing was positive. All the players learned the game mechanics before the end of the game and they beat the challenge. However, some of the players had a hard time in learning how to use and switch weapons. This would be improved by small changes to the UI layout and to how the buttons look. Also, it is assumed that a short tutorial about weapons would have fixed the misunderstandings.

4.10 First PvP user testing

After the landscape testing, one modification was made to the game before the new player-versus-player (PvP) test session was held. PvP stands for player versus player. In the game it was implemented only in the form of online battle.

The only modification was concerning about the weapon switching. The primary weapon button and the secondary weapon button were placed on the same button, making a single button to alternate between the primary and secondary weapons. The image of the new UI can be seen in **Figure 36**. When the player pressed the new switch-weapon button, the switching of the gun was shown in the button by an animation were the two weapon images switched places, having the active weapon image on top of the other weapon image.

Figure 36. In the new UI, the two weapon-switching buttons were merged. Changing weapon between the primary and secondary weapon happened by the same button. The active weapon at the moment is the sniper.

4.10.1 Setup

After having a positive conclusion for the learnability and understandability of the UI, a new play session was held for getting more results on the UI and more importantly, more realistic results about the UX. For the UX part, the team wanted to know whether or not the players enjoyed and had fun during the gameplay.

Before the starting the game, the testers were informed about what the team were going to test. The test was about online battles against teams and how the players learned the game functionalities, and whether they enjoyed the game. The testers were informed about the session length, which was about 30 minutes. This included setting up the test devices, an approximately 15 minutes long gameplay session, and finally a questionnaire at the end. The testers played the game with different mobile devices from phones to tablets including both Androids and iPhones.

4.10.2 Play session

Nine players were invited for the play session internally from Rovio. The testers were playing the game 15 minutes as a group, allowing discuss during the play session. The 15 minutes of gameplay was equal to two matches in a 3-vs-3 game mode. As there were not enough players to form two simultaneously running games with only the testers, three persons from the developing team joined the play session. After the play session, the

testers were asked to fill a questionnaire about their opinions of the UI and UX during gameplay.

4.10.3 Results

Answering questions 1-6 (below) was done by selecting one of the following options: (1) negative, (2) slightly negative, (3) neutral, (4) slightly positive, and (5) positive. For the question number 7 (below), answering was done by giving a number between 1 and 10 where 1 means poor and 10 means great. Also, open feedback for combats and for the game as overall was given by the questionnaire.

The following represents the questions and the answers. Answers per question were divided into three parts.

1. How would you rate your initial reaction towards the game following this playtest session?
   - Average: 4.3
   - Lowest: 3
   - Highest: 5

2. Did you enjoy the character(s) that you played?
   - Average: 3.3
   - Lowest: 3
   - Highest: 5

3. Would you play the game again?
   - Average: 4.4
   - Lowest: 3
   - Highest: 5

4. Would you look forward to trying out new characters and weapons?
   - Average: 4.3
   - Lowest: 3
   - Highest: 5

5. Did you understand the game controls?
   - Average: 2.7
   - Lowest: 2
   - Highest: 5

6. Did you enjoy the feel of the combats?
   - Average: 3.7
   - Lowest: 3
   - Highest: 5

7. How would you rate your overall enjoyment of this version of the game?
   - Average: 7
   - Lowest: 6
   - Highest: 8

In addition, 33% of the testers found the game to be fun and they liked how important it is to play as a team. Some players gave positive feedback about easy movement and that the controls were understandable and easy to use. However, the majority of the players did not understand the controls as well as the others or that the controls felt unintuitive for them. Other positive feedback towards combat and the game overall was hectic gameplay and the variety of skills.

Multiple aspects were found to give a negative user experience during combats. One feature that was mentioned most often was the difficulty with aiming. One suggestion to get controls better was to switch the weapon back to the default weapon after using the secondary weapon. This way, the players should not have to keep thinking which weapon is the active one at the moment. 78% of the testers had difficulties with using weapons. Switching weapons was hard at the first but it was learned during the first game. Aiming was seen as hard, especially with high arc skills, or the testers felt that the shooting was not fully responsive and mixing automatic and manual aiming was seen confusing for some players.

4.10.4 Analysis of the results

From the questions 1-4, it can be seen that the players were positive about the game and they would like to play the game again. The answers varied between 3 and 5 with the average answer being 4.4. This means that the players had a good user experience during overall gameplay. Even though the players did not have much time for learning the UI and game mechanics nor could they learn them in peace without rushing, they had a positive feeling towards the game. The initial reactions towards the game were between 3 and 5 with the average being 4.3. This is an important phase to make the first impression for players and, as a prototype game, 4.3 was seen as a great result.

Enjoyment towards the playable characters varied from 3 to 5 with an average of 3.3. The majority rating it a 3 may have come from misunderstanding the character's skills and how to use them. Also, lacking the many external differences between the characters may have made them less interesting or unique. The high average of 4.3 of looking forward to new characters and weapons showed that the game was at least somewhat interesting and that they would like to try new actions in the game.

This information was very important for the game developing progress. This gave the developer team the first real unbiased data of how the gameplay feels during online PvP matches. As the results were positive, the overall feeling and structure of the game could have been seen as working, although new characters and skills were needed along with game balancing.

Even though the players did not fully understand the game controls, they were still slightly positive about combats during gameplay. The answers for the game controls varied from 2 to 5 with the average being at 2.7. Although the controls were clear for some players, the majority of the testers had slightly negative experiences with learning the controls. The main problems in understanding game controls were the same as in the previous landscape user testing. It took time for players to understand how weapon switching worked and what were the ways of aiming. Using the high arc skill was hard for some players. The difficulties for using the skill can be the small joystick area in the high arc skill button. As the testers played the game with different mobile devices, the aiming

joystick area with the high arc skill could have been very small, with some devices making aiming with it very hard. Some players preferred manual aiming whereas other players preferred automatic aiming. However, in addition to adjusting the feeling of aiming, the main problem of understanding how to use different buttons was assumed to be fixable with a short tutorial before the first online match.

Even though the controls were hard to learn, the players enjoyed combat in the game. The results for combat feel varied between 3 and 5 with an average of 3.7.

The answers for overall enjoyment varied from 6 to 8 with an average of 7. To improve gameplay, the controls have to be made clearer for the players. One way could be to provide a tutorial when launching the game for the first time. Then the players would be guided to use and understand the UI in the correct way. Another way is to modify the UI and actions to be more similar to the other games in the same genre.

Even though the tester group size was quite small and it was not large enough to get very accurate data as in a large data set, it gave the team the necessary information on the gameplay that pointed out the best and worst features in the game.

4.11 Final PvP user testing

After the first PvP user testing, the author and his team decided to try how the game would work and how the controls would be understood if the weapon switching would be disabled. In this version, the controls would be similar to the other described MOBA games in chapter 3.5. Players would always have a default attack and three skills to be used. The skills would be aimed from the skill button itself instead of from the old ‘shoot button’. The ‘shoot button’ would be changed to be the default attack, which would correspond to the primary weapon. However, the charging rules for the ultimate skill would stay the same. The new look of the UI can be seen in **Figure 37**.

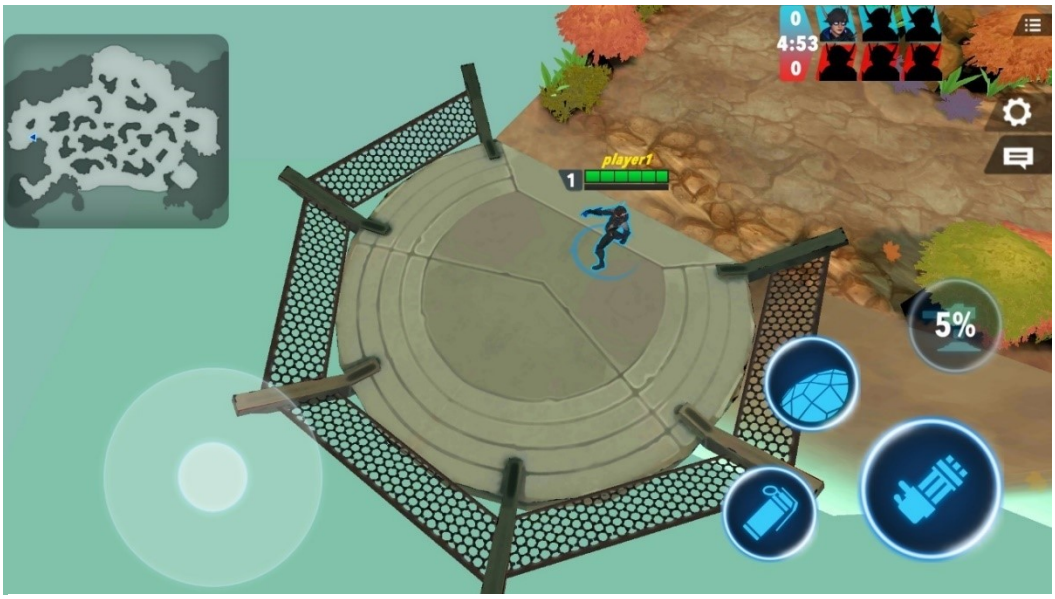

Figure 37. The new UI. All skills are used and aimed from the respective buttons.

Two new user test sessions were held with the same pipeline in the test sessions. The first play session used the weapon-switching build and the second testing used the new UI.

These test sessions were held to gather data for whether the game controls would be clearer to the players or not in the new UI than in the weapon-switching UI. The team also wanted to gather more data on whether the players enjoyed and had fun during the gameplay or not.

4.11.1 Setup

For both play sessions, six testers were invited to play the game as a group in the same room allowing discuss during the whole test session. The testers were different for different UIs. The testers played the game with different mobile devices from phones to tablets including both Androids and iPhones. The sessions were observed by an observer who was in the play session room with the testers. Other team members were watching the play sessions via video stream.

4.11.2 Play session

After an introduction to the game and what the test session would contain, the testers played three matches lasting about 20 minutes. This was followed by a questionnaire. This was done for both play sessions.

4.11.3 Results

8 out of 10 questions were answered with numbers between 1 and 4, where 1 is strongly disagree and 4 is strongly agree. The average answers to these questions from both play sessions are visualized in **Table 1**. Test 1 results were for the weapon-switching build and test 2 results were for the newest UI without weapon switching. The other two questions were “how you would improve or change the controls” and “what would be needed to improve with user interface”.

Table 1. Results of the play sessions.

Question	**Test 1 average**	**Test 2 average**
Controls were easy	3	3.2
Moving was enjoyable	3.2	3
Easy to change weapons	2.7	-
Skills with cooldowns were easy to use	1.3	3
Automatic aiming worked well	2.8	2
Manual aiming was easy	2	2.5
Ammunition was clear	1.3	-
Easy to understand when shooting is enabled	2.8	2.7
Total of all questions	2.4	2.7

4.11.4 Analysis of the results

In both builds, the players found the movement to be enjoyable and working. Also, the controls were also seen to be easy to learn overall. Both of these questions had an average score of at least 3.

In the weapon-switching build, both unique aspects, which were weapon switching and the ammunition, were not fully understood by all players. As in the two previous play sessions, the players were confused at the beginning about how the UI buttons worked, how the weapon was changed, and which buttons were meant for certain actions such as aiming. Even though most of the players eventually learned how the UI functionalities worked, some players still did not find these intuitive. The amount of ammunition and how it affected the gameplay was clear only to one of the players. Other five players either did not know about its existence or it was not clear enough for the players. The existence of ammunition was probably ignored by the fact that the weapons also had cooldown system. After shooting with a weapon, a cooldown was started as with other skills. However, after ammunition ran empty, the reload started which was very similar to the regular cooldown. The reload cooldown had a slightly different length and a reload icon over the player. This result confirmed the team that the ammunition in the gameplay is not necessary and it could be overridden with a regular cooldown indicator.

Unlike in the new UI, in the weapon-switching build, only one skill had a regular cooldown. In the weapon-switching build, the players had problems with learning how to use the skill properly, therefore having the average score of 1.3. After changing all skills to be cooldown based (except the ultimate skill, which still had to be charged), the players were able to learn and use them better and therefore have the average score of 3. From these results, it can be seen that having multiple differently working buttons that looked somewhat similar had a huge negative effect on how players learned the controls. Therefore, this new UI seemed to work much better in terms of the level of understanding and consistency.

Manual aiming got slightly negative results on both builds. One major factor for this was that some of the players did not use manual aiming even once. The answer with these players was marked as 1 because the manual aiming was not seen as easy if the players did not realize its existence after playing three matches. If the answers for this question for these players would have been ignored, the results for test 1 and test 2 would have been 2.2 and 2.8, instead of 2 and 2.5 respectively. After pressing the skill button or ‘shoot button’ in the weapon-switching build, only the maximum range circle was visible for the players, indicating how far the player can target enemies. Starting to drag a finger while aiming automatically to start manual aiming was not clear enough. For future updates, a joystick should be drawn over a button that is being pressed to indicate to players that they can move the finger inside the joystick and that something would happen. This would probably encourage players to try to move the finger, after which they would understand how manual aiming works.

The results for automatic aiming for test 1 and test 2 got average scores of 2.8 and 2 respectively. One of the players in test 2 did not use automatic aiming at all. This answer was marked as 1 because the automatic aiming was not clear to the player even after three matches. If the answer by this player would have been ignored, the average score for test

2 would have been 2.2. The lower score results for automatic aiming could have come from not fully understanding automatic aiming rules. While attacking against several enemies, it may not be clear to the player that the enemy with lowest amount of HP will be selected as the target, even if there would be closer targets. Also, in the new UI, the players were using skills more often accidentally and therefore having to wait to get the skill to be active again.

The average score of understanding when shooting was enabled was almost the same in both builds. Cooldown indicators in the skill buttons and in the 'shoot button' were not always clear to the players or they did not see it under their fingers. This happened often in the weapon-switching build where players were rapidly pressing and releasing the 'shoot button' and at the same time blocking the view to the cooldown indicator. Also, charging of the ultimate skill was not clear to the players at the beginning of play sessions. A common mistake was to think that the percentages in the button indicates how powerful the shot would be. In this case, the players tried to start using the ultimate skill without success before the charge was completed. One possible fix to misunderstanding the cooldowns would be changing the button color to grey and slightly transparent when the button cannot be used. This could better indicate whether a button can be used or not.

The answers to how the players would improve or change the controls and what would be needed to improve in the UI are covered next. The following answers were given in the weapon-switching build. One suggestion was to enable shooting while holding a finger in the 'shoot button'. This would remove the need to continuously tapping the 'shoot button'. This could also work in the new UI with the default attack because of its short cooldown time. Also, it was wanted that the ultimate skill could be aimed from the ultimate button itself, thus making it work as a skill rather than as weapon-switching button. Two of the players wanted clearer cooldown feedback. A possible solution for this has been covered above. The following suggestions were made for improving UI: buttons that cannot be used should be grey, one's own hero should standout better from the group of players in the gameplay, and the minimap should be bigger. The player's own hero may standout better by having something unique UI aspect with own hero, for example. This could be a different color in the health bar (green for own hero, blue for friendlies, and red for enemies), an indicator below or above the hero, or by having a glow for one's own hero. The minimap was seen to be too small for some players with smaller devices. In this case, it was hard to recognize where the players were in the minimap.

In the new UI it was again suggested that the way cooldowns were shown should be improved. One player mentioned accidentally activating skills by tapping and releasing a skill button without targets. Using skills with automatic aiming without targets should be disabled to prevent players from accidentally using skills and therefore possibly making them an underdog in combat. One of the players would not change anything. Two of the players saw the minimap as too small and difficult to read properly. One of the players did not notice team scores or timer well, and it caused some misunderstanding of the state of play. Furthermore, more visible feedback from an action, attacking, and taking damage was wanted.

5 Conclusion

The purpose of this master's thesis was to design and develop a UI for a MOBA mobile game. The main aim was to create the game in portrait mode and to facilitate easy-to-understand and easy-to-use one-handed gameplay. The secondary goal was to make the game work in landscape mode with two-handed gameplay if development of the portrait mode version was not successful.

This master's thesis had two key conclusions. The first conclusion was that general MOBA mobile games are hard to transfer into portrait mode without making major changes in game rules and user interface. Secondly, rules and UI have to be simple and easy to understand in order to help players learn the game mechanics. Even small changes to an existing working UI may make the UI incomprehensible.

To make the game playable with one hand, only one action could be performed at a time to prevent any advantages in two-handed gameplay. Using different joysticks for moving and shooting was considered to be a bad implementation. During changes between actions, there were small delays that players tried to avoid by using both hands. Whether players played the game with one or two hands, they did not like the break in the gameplay that was caused by the casting time in skills. One potential way to create a working one-handed gameplay for portrait mode could be the addition of the UI features of tap-to-move and tap-to-attack. However, this possibility wasn't tried by the team because it could have had negative effects to the game feel and immersion. Removing the casting time from the gameplay would make both the shooting phase and switching between actions (i.e. movement and shooting) faster and thus allowing a greater advantage for playing with two hands. At this point, it did not seem intuitive to prevent players from moving and shooting at the same time. Furthermore, as seen in the results of the portrait-mode user testing, fingers blocked many parts of the UI controls and game view, which made the user experience uncomfortable. Due to these reasons, the portrait mode was seen as a failure and the game was changed to work in landscape mode with two-handed gameplay.

Even though the portrait mode of the game was a failure, it does not mean that enjoyable one-hand MOBA games for portrait mode cannot be made. The author is of the opinion that the portrait mode version failed for two main reasons. First, the team wanted the game experience to feel the same as in other successful landscape-mode MOBA games but to retain the portrait mode orientation as in the original Brawl Stars. This created problems because all games that were referred to were played with two fingers except Vainglory, where the two-hand gameplay was only highly recommended. Having a one-handed gameplay tends to limit the players to only one action at a time and that may create a major difference in terms of enjoyability. By removing the functionality of doing simultaneous actions in a game without a good reason may negatively affect game experience. If the MOBA games referred to were to allow the players only to move or attack, most likely the whole gameplay would be affected. In the case of Vainglory, this would probably decrease the speed of changing actions and the UI layout would have to be redesigned for ergonomic one-hand play.

The second reason for the failure of the portrait mode was that the team did not want to abandon direct control of the hero character. By changing the game mechanics to tapping,

there might be a possibility that a player could relate more as the commander of the hero rather than experiencing oneself to be the hero. Nevertheless, the tapping mechanism does not require two-handed gameplay. However, continuous tapping on portrait mode could easily block some important parts of the UI. Furthermore, the portrait mode game could have been fixed to be more enjoyable by changing the game mechanics to tap-to-action instead of using a virtual joystick. Therefore, a one-handed portrait MOBA game that is played by tapping seems to be a good option for creating a successful game.

After switching from portrait to landscape mode, user testing results indicated an improved gameplay experience. The weapon-switching build made the game clearer, more intuitive, and more enjoyable. Also, although the landscape weapons-switching build had a more complex UI to operate than the portrait mode one, the players were able to learn the mechanic without tutorials albeit with delay and difficulty. In spite of this, tutorials would still be necessary to ensure that all players learn the controls correctly.

The UI was considered successful if test users were able to start playing without any instructions in the beginning or middle of gameplay and they did not need any help in making actions or using the interface. Also, players would have to find the game enjoyable. With these goals and results in mind, the project for designing a UI for a mobile MOBA game can be seen as finished successfully. Furthermore, the game was made more intuitive and even faster to learn by removing the weapon-switching mechanism and using the skills in the same way as in the other common MOBA games.

The latest user testing indicated that the players were able to learn the actions faster without misusing them when the skills were used directly from the skill buttons itself. If the weapon switching had been left in the gameplay, several updates would have been needed. By making the UI elements more distinct from one another, players would easily differentiate between differently behaving button groups. This means, for example, that the ability buttons should have a different appearance from weapon-switching buttons. Similarly, the shoot button should also have a unique appearance. Furthermore, a tutorial for teaching the basic usage of the UI should be developed to prevent players from misunderstanding the UI and game mechanics.

It is not coincidental that most of the MOBA games listed in this master thesis have several common features: orientation of the game, UI appearance, and the gameplay mechanics of moving and using attacks. As seen in this master's thesis, having the core gameplay features and mechanics similar to the other MOBA games makes the game faster to learn and understand by new players. Changing the shooting rules to having always one active weapon or trying to make MOBA mobile game be played single handed in portrait mode has its own problems in terms of understandability and readability without any kind of tutorials. The last version of the game that was developed by the author and his team was seen as success in terms of UI and UX design.

In general, while designing and developing a game, it is important to keep the gameplay UI simple. Having a complex UI or a UI full of differently behaving buttons can make players confused and therefore either misunderstand the mechanics of the UI or that players will not try to use all the features. Furthermore, if a game genre has a UI layout that has been commonly used in games, having a similar UI in the developed game may be a good idea if it fits with the game mechanics. Having commonly used UI layout in a

game will help players from that genre to increase the speed of understanding the game by learning the UI and adopting the game features faster.

In the general case of game design and development, transferring a two-handed landscape game to work in portrait mode as a single-handed game will presumably need changes in the gameplay mechanics. Firstly, a major change is to restrict the simultaneous use of two fingers or at least not provide any advantages to using two fingers. If the game gives players advantages by using two fingers simultaneously, the game should be played with two fingers regardless of the orientation of the game. However, changing the game to work with only one finger may have effects to the game flow and rules, therefore having effects on user experience. In this case, the game design should be started without having multiple main core features and actions directly from the landscape version. The project that the author was developing had this very problem. The need to have features, actions, and a game feel very similar to other landscape MOBA games made the game take a development path that was the wrong direction and made the result unenjoyable.

The results of this master's thesis can be utilized as references for do's and don'ts in designing and developing real-time action based mobile games.

References

1. Llanos SC, Jørgensen K. Do players prefer integrated user interfaces? A qualitative study of game UI design issues. 2011.

2. Galloway AR. Gaming essays of algorithmic culture. https://www.invisionapp.com/blog/design-for-iphone-x/ . Updated 2017. Accessed 1/4, 2018.

3. Iacovides I, Cox A, Kennedy R, Cairns P, Jennett C. Removing the HUD: The impact of non-diegetic game elements and expertise on player involvement. 2015.

4. Tach D. Deliberately diegetic: Dead space's lead interface designer chronicles the UI's evolution at GDC. https://www.polygon.com/2013/3/31/4166250/dead-space-user-interface-gdc-2013 . Updated 2013. Accessed 2/5, 2018.

5. Chen J. Designing journey. http://www.gdcvault.com/play/1017700/Designing . Updated 2013. Accessed 2/14, 2018.

6. Todaro JA. 7 ways to design for the new iPhone X. https://www.invisionapp.com/blog/design-for-iphone-x/ . Updated 2017. Accessed 1/4, 2018.

7. Tencent Games. Arena of Valor: 5v5 Arena Game. 2017; 1.19.1.1.

8. Cabbiegames. Classic Labyrinth 3d Maze. 2018; 5.5.

9. Halfbrick Studios. Fruit Ninja. 2017; 2.5.9.471383.

10. Perfect Tap Games. Chicken Scream. 2017; 1.5.1.

11. Takatalo J, Häkkinen J, Kaistinen J, Nyman G. User experience in digital games. 2008.

12. Ryan RM, Rigby CS, Przybylski A. The motivational pull of video games: A self-determination theory approach. *Motiv Emotion*. 2006;30(4):347-363.

13. Hämäläinen P, Takatalo J. Millainen peli koukuttaa ja tuottaa mielihyvää? *Duodecim*. 2017;23.

14. Hunicke R, LeBlanc M, Zubek R. MDA: A formal approach to game design and game research. 2004.

15. Lazzaro N. Why we play games: Four keys to more emotion in player experiences. 2004.

16. Silvia PJ. Interest—The curious emotion. *Curr Dir Psychol Sci*. 2008;17(1):57-60.

17. Long S. What is games ‘User experience’ (UX) and how does it help? https://www.gamasutra.com/blogs/SebastianLong/20171002/306649/What_Is_Games_User_Experience_UX_and_How_Does_It_Help.php . Updated 2017. Accessed 5/12, 2017.

18. Silva VdN, Chaimowicz L. MOBA: A new arena for game AI. 2017.

19. Yang P, Harrison B, Roberts DL. Identifying patterns in combat that are predictive of success in MOBA games. 2014.

20. Linn D. Most played games: November 2015 – fallout 4 and black ops iii arise while starcraft ii shines. http://caas.raptr.com/most-played-games-november-2015-fallout-4-and-black-ops-iii-arise-while-starcraft-ii-shines/ . Updated 2015. Accessed 11/28, 2017.

21. Riot Games. League of Legends. 2009.

22. Valve Corporation. Dota 2. 2013.

23. Gaudiosi J. Riot games' league of legends officially becomes most played PC game in the world. https://www.forbes.com/sites/johngaudiosi/2012/07/11/riot-games-league-of-legends-officially-becomes-most-played-pc-game-in-the-world/#43c82af4718b . Updated 2012. Accessed 11/28, 2017.

24. Bulatov V. Arena of valor - true MOBA on mobile phones. https://www.gamasutra.com/blogs/VadimBulatov/20171226/312233/Arena_of_Valor__true_MOBA_on_mobile_phones.php . Updated 2017. Accessed 01/06, 2018.

25. Bergstrom-Lehtovirta J, Oulasvirta A. Modeling the functional area of the thumb on mobile touchscreen surfaces. 2014.

26. Swink S. *Game feel: A game designer’s guide to virtual sensation.* Elsevier; 2009.

27. Jacob RJK. The future of input devices. *ACM Computing Surveys (CSUR)*. 1996;28(4es).

28. Fullerton T. *GAME DESIGN WORKSHOP: A playcentric approach to creating innovative games.* 2nd ed. Elsevier Inc.; 2008.

29. Gaea Mobile. Ace of Arenas. 2017; 2.0.8.0.

30. Gameloft. Heroes of Order & Chaos. 2017; 3.6.1e.

31. Moonton. Mobile Legends: Bang Bang. 2018; 1.2.48.2451.

32. Reality Squared Games. Heroes Evolved. 2018; 1.1.23.0.

33. Super Evil Megacorp. Vainglory. 2018; 2.12.0.

34. Supercell. Brawl Stars. 2018; 6.52.

35. uCool. Heroes Arena. 2018; 1.4.3.

36. Why can’t I find brawl stars in my app store or the google play store? https://supercell.helpshift.com/a/brawl-stars/?s=hot-topics&f=why-can-t-i-find-brawl-stars-in-my-app-store-or-the-google-play-store&l=en&p=web . Updated 2017. Accessed 2/15, 2018.

37. Supercell. Brawl Stars. 2018; 7.278.